cercetare operațională cu aplicații în economie

Scanned with OKEN Scanner

MINISTERUL EDUCAȚIEI ȘI ÎNVĂȚĂMÎNTULUI

Conf. dr. ing. Gheorghe Boldur — Lățescu
Conf. dr. Ion Săcuiu
Conf. dr. Eugen Țigănescu

cercetare operațională cu aplicații în economie

Editura didactică și pedagogică — București, 1979

Scanned with OKEN Scanner

Cuprins

1. INTRODUCERE

1.1. Cercetarea operațională și disciplinele înrudite 5
1.2. Rolul modelării în cercetarea operațională 10

2. PROGRAMAREA LINIARĂ

2.1. Considerații generale privind modelul matematic al problemei de programare liniară 15
2.2. Forme de prezentare a modelului matematic 22
2.3. Teoreme fundamentale 25
2.4. Interpretarea problemelor de programare liniară 28
2.5. Algoritmul Simplex 31
2.6. Convergența algoritmului simplex. Degenerarea în programarea liniară . 41
2.7. Determinarea unui program de bază inițial. Metoda celor două faze. Metoda bazei artificiale 44
2.8. Problema duală 52
2.9. Alte metode de rezolvare a problemelor de programare liniară 63
2.10. Reoptimizarea și parametrizarea în programarea liniară 74
2.11. Probleme de programare liniară de tip transport 97

3. TEORIA GRAFELOR

3.1. Obiectul teoriei grafelor 117
3.2. Concepte fundamentale 119
3.3. Concepte neorientate 123
3.4. Determinarea drumului de valoare optimă 124
3.5. Rețele de transport 131
3.6. Drumuri și circuite hamiltoniene 134

4. ANALIZA DRUMULUI CRITIC

4.1. Trecerea de la metode tradiționale de planificare a acțiunilor complexe la metodele A.D.C. 142
4.2. Metoda C.P.M. 146
4.3. Grafice A.D.C. integrate și condensate 152
4.4. Actualizarea graficelor A.D.C. 155
4.5. Optimizări cost-durată 156
4.6. Analiza resurselor 161
4.7. Alte procedee de analiză a drumului critic 172

Scanned with OKEN Scanner

4.8. Procedee complexe de alocare-nivelare a resurselor: sisteme de programe pentru calculator 76

5. GESTIUNEA STOCURILOR

5.1. Introducere 78
5.2. Reguli de gestiune 79
5.3. Modele matematice deterministe folosite în optimizarea proceselor de stocare 82
5.4. Model de gestiune cu perioadă fixă şi cerere aleatoare cu posibilitatea de rupere a stocului 195
5.5. Metode euristice de gestiune a stocurilor 202

6. PROGRAMAREA DINAMICĂ

6.1. Introducere 203
6.2. Modelul matematic al programelor dinamice discrete ce se desfăşoară în viitor sigur şi cu orizont limitat 203
6.3. Programe dinamice staţionare 204
6.4. Model dinamic de programare optimală a investiţiilor 218
6.5. Interpretări ale programelor dinamice cu ajutorul teoriei grafelor. Grafe secvenţiale, ... 221
6.6. Programe dinamice discrete în viitor sigur şi orizont nelimitat 224
6.7. Criteriul valorii actuale 226
6.8. Criteriul valorii medii pe o perioadă 228

7. ELEMENTE DE TEORIA AŞTEPTĂRII

7.1. Noţiuni introductive 230
7.2. Elemente de bază într-un model de aşteptare 230
7.3. Modele de aşteptare cu o singură staţie de serviciu 232
7.4. Modele de aşteptare cu mai multe staţii de serviciu . . . . . .

8. TEORIA JOCURILOR ŞI A DECIZIEI

8.1. Elemente privind jocurile strategice 269
8.2. Teoria deciziei 272
8.3. Clasament şi alegere după criterii multiple 281
8.4. Programarea liniară cu mai multe funcţii obiectiv 287

9. PROGRAMAREA OPERATIVĂ A PRODUCŢIEI

9.1. Aspecte generale 297
9.2. Stabilirea ritmurilor de fabricaţie în condiţiile executării unui plan realizabil la nivelul resurselor efectiv disponibile 300
9.3. Stabilirea ritmurilor de fabricaţie cu ajutorul tehnicilor de dinamică industrială 301

10. BIBLIOGRAFIE

Scanned with OKEN Scanner

Capitolul I

INTRODUCERE

1.1. Cercetarea operațională și disciplinele înrudite

Cercetarea operațională este una din disciplinele care a apărut către sfîrșitul primei jumătăți a secolului nostru și s-a dezvoltat spectaculos în special în ultimii 20—25 de ani, în strînsă legătură cu o serie de alte discipline ale organizării și conducerii, cum ar fi cibernetica, informatica, sau analiza sistemelor, toate împreună fiind produse ale revoluției tehnico-științifice contemporane.

Pentru a avea o imagine de ansamblu asupra obiectului cercetării operaționale aplicate în economie, considerăm deci, util să examinăm succint cum au apărut și au evoluat disciplinele organizării și conducerii precum și legăturile pe care le prezintă între ele.

Concepția „organizării științifice", conturată către sfîrșitul secolului al 19-lea și începuturile celui actual, consideră unitatea productivă ca un mecanism, în care oamenii ajutați de mașini lucrează într-un determinism aproape total, pe baza unor dispoziții acționînd ierarhic, conform unor competențe riguros definite.

Principalii reprezentanți ai începuturilor organizării științifice, care formează așa-numita „școală clasică", stabilesc pentru prima oară o serie de principii ale conducerii științifice. Printre acestea figurează binecunoscutul (și încă actualul) principiu al excepției, principiul specializării organizaționale, principiul definirii riguroase a sarcinilor, principiul organizării ierarhice (Staff and Line) ș.a.

Între conceptele utilizate de școala clasică nu figurează informația și nici decizia : conducerea „mecanismului" economico-social revine (în ultimă instanță prin parcurgerea treptelor piramidei ierarhice), întotdeauna, unui centru unic de decizie, pentru care informațiile sînt presupuse, aprioric, disponibile complet și instantaneu fără nici un fel de restricție (de timp, de spațiu, de tehnică a transmiterii și înmagazinării etc.).

Cu toată perspectiva sa limitată, școala clasică are marele merit al desțelenirii unui domeniu virgin. Pionierii organizării științifice (Taylor, Gantt, Fayol) și ceilalți reprezentanți ai școlii clasice pun pentru prima oară problema abordării raționale a mecanismului funcționării unei întreprinderi.

Scanned with OKEN Scanner

O mare parte din ideile școlii clasice au fost criticate de reprezentanții diferitelor școli care s-au dezvoltat ulterior în științele organizării, dînd naștere, după cum vom arăta în continuare, unor teorii din ce în ce mai abstracte și mai complexe. Merită să arătăm că în deceniul al șaselea, ca o reacție împotriva excesului de teoretizare, s-a dezvoltat o așa-numită școală neoclasică, avînd drept obiectiv reîntoarcerea la practică.

În deceniile care urmează după apariția și dezvoltarea școlii clasice, problemele informațional-decizionale își afirmă prezența din ce în ce mai acut, pe măsura creșterii dimensiunilor și complexității organizațiilor social-economice și își caută rezolvări empirice de cele mai multe ori nu la nivelul necesităților. Se stabilesc adesea circuite informaționale paralele și supraabundente (redundante) iar în afara fluxurilor oficiale (formale) de date, se dezvoltă o circulație neformală, uneori mai eficientă dar cu caracter strict local. În procesele de decizie continuă să prevaleze rutina, bunul simț, talentul sau chiar improvizația.

În perioada următoare primului război mondial au putut fi constatate ca urmare a acestor rezolvări empirice diferențe considerabile, din punctul de vedere al competitivității, între unități economice cu structuri organizatorice și dotări tehnice identice sau similare. Analizele efectuate au condus la o primă includere în perimetrul cercetării privind problemele organizării și conducerii a aspectelor informațional-decizionale, pînă atunci ignorate, și totodată și a aspectelor relațiilor umane. Se lărgește considerabil problematica organizării și conducerii și încep să circule cu din ce în ce mai multă autoritate denumirile de management (ca activitate practică) și management science (știința conducerii).

Această perioadă este dominată de „școala comportamentului" care pune în centrul preocupărilor sale observația minuțioasă a comportamentului oamenilor în timpul procesului productiv și căutarea motivațiilor care determină coeziunea grupurilor.

Diferențele substanțiale între școala comportamentului și școala clasică se referă în special la aspecte ca : descentralizarea deciziilor, promovarea încrederii între membrii unui grup (și neglijarea autorității) cu accentul pus pe responsabilitate (și nu pe control)*.

Începînd cu deceniul al cincilea al secolului nostru**, se produce un fenomen care promovează informația și decizia printre elementele esențiale ale epocii în care trăim.

La acest fenomen contribuie în primul rînd creșterea extraordinară a complexității, structurale și funcționale, a organizațiilor economice. Procesele de comasare-integrare, apariția structurilor organizaționale cu activități productive pe arii geografice foarte mari (și, de asemenea, cu multiple probleme legate de desfacerea produselor), ridicarea nivelului de tehnicitate a instalațiilor și corespunzător o specializare accentuată a profesiunilor — sînt numai cîteva din aspectele acestei complexități a unităților productive moderne.

---

* Printre reprezentanții „școlii comportamentului" pot fi citați Mayo, Abraham Zalesnick și D. C. Peltz.

** Desigur, referirea la un moment în timp nu poate fi decît pur orientativă ; aici am avut în vedere apariția primei generații de calculatoare electronice, a primelor lucrări de cibernetică și a primelor echipe de cercetare operațională.

Scanned with OKEN Scanner

Ca o consecinţă a acestei stări de fapt apare o extraordinară creştere a cantităţii de informaţii, deţinute şi manipulate în unităţile productive, accentuată şi de formularea unor condiţii mult mai severe în ceea ce priveşte calitatea informaţiei (pertinenţa şi operativitatea acesteia). Alături de producţia de bunuri apare o producţie de informaţii din ce în ce mai însemnată, informaţia devine chiar un produs sau marfă ce se poate negocia, ajungînd alături de servicii, obiectiv al unor organizaţii specializate.

În ceea ce priveşte procesele de decizie, pentru prima oară se pune în mod riguros şi pe scară largă problema găsirii unor soluţii optime sau apropiate de cele optime, în marea diversitate de probleme organizatorice şi de conducere.

Se poate considera că toate aceste schimbări au condus la o veritabilă revoluţie informaţional-decizională în domeniul organizării şi conducerii şi, ca o consecinţă, la apariţia managementului ştiinţific modern.

Principalele discipline privind conducerea, care au apărut în această etapă, sînt: cercetarea operaţională, cibernetica, informatica, psihosociologia organizării şi teoria generală a sistemelor.

● **Cercetarea operaţională,** care poate fi definită succint ca disciplină a optimizării deciziilor cu ajutorul modelării matematice, a apărut în perioada celui de-al doilea război mondial.

Considerată de unii ca reprezentînd *şcoala matematică* în disciplinele organizării şi conducerii, cercetarea operaţională se caracterizează în primul rînd prin procesul de elaborare a modelelor, de regulă matematizate, care descriu procesele economice pentru care urmează a se lua decizii cît mai avantajoase. Avînd în vedere importanţa modelării în cercetarea operaţională, îi vom consacra în întregime paragraful următor.

● **Cibernetica** este ştiinţa care se ocupă cu conducerea şi reglarea sistemelor complexe.

Printre încercările cele mai caracteristice de perfecţionare a metodelor folosite în ultimele decenii în ştiinţele organizării şi conducerii, alături de întrebuinţarea masivă a procedeelor matematice şi a calculatoarelor electronice, se află şi utilizarea concepţiei sistemico-cibernetice.

Poate fi definită ca sistem, orice secţiune a realităţii în care se identifică un ansamblu de fenomene, obiecte, procese, concepte, fiinţe sau grupuri interconectate printr-o mulţime de relaţii reciproce, precum şi cu mediul înconjurător şi care acţionează în comun în vederea realizării unor obiective bine definite. Mulţimea elementelor şi a relaţiilor dintre acestea, precum şi a relaţiilor între componente şi ansamblu formează structura sistemului. Mulţimea caracteristicilor unui sistem, la un moment dat, determină starea sa.

Pentru analiza comportamentului sistemelor, în ansamblul lor, s-a propus conceptul de „cutie neagră" care reprezintă sistemul privit ca un tot, făcînd abstracţie de procesele sale interne. Cutia neagră primeşte impulsuri din partea mediului înconjurător („intrările" în sistem) şi, prelucrînd aceste impulsuri, le transformă în acţiuni asupra mediului („ieşirile") din sistem.

Sistemele se pot clasifica: după natura lor (sisteme naturale — cum sînt organismele vii şi sisteme elaborate — tehnice, economice, concep-

tuale), după modul de funcţionare (deschise — în care ieşirile nu influenţează intrările, şi închise, în care are loc influenţa intrărilor de către ieşiri) şi după comportament (deterministe sau probabilistice*).

Mecanismul transformărilor intrărilor în ieşiri poate fi descris cu ajutorul funcţiilor de transfer care au diverse forme particulare, după natura sistemului.

Sistemul devine cibernetic** atunci cînd apare reglarea (conexiunea inversă, feedback), adică o intervenţie asupra intrărilor în scopul menţinerii ieşirilor la nivelul unor parametri-obiectiv doriţi.

Se înţelege că expresia analitică a funcţiilor de transfer şi a mecanismului reglării conduce la forme matematice foarte diverse şi de cele mai multe ori foarte complexe.

Ansamblul economiei poate fi privit ca un sistem ale cărui elemente componente (organizaţiile social-economice de diferite mărimi) sînt intercorelate prin fluxuri materiale şi informaţionale şi au un comportament orientat spre atingerea unor obiective precise. La rîndul lor, organizaţiile, care sînt elemente componente ale sistemului-ansamblu, pot fi considerate sisteme, diviziunea putîndu-se continua pînă la identificarea unor componente elementare indivizibile. Scopul cercetării cibernetico-sistemice aplicată la realitatea social-economică îl constituie surprinderea comportamentului sistemelor, una din căile de descriere a acestui comportament fiind găsirea expresiei funcţiilor de transfer şi a mecanismului reglării.

Adoptarea perspectivei cibernetico-economice în ştiinţele social-economice reprezintă un cîştig teoretic remarcabil şi este foarte probabil ca, în următorii ani, să asistăm la închegarea unei teorii cibernetico-sistemice complete şi unitare aplicată la realitatea social-economică pe scară largă. Este de remarcat contribuţia economiştilor din ţările socialiste la naşterea şi dezvoltarea ciberneticii economice***, la loc de cinste situîndu-se lucrările şcolii româneşti de cibernetică economică condusă de acad. Manea Mănescu.

● **Informatica** poate fi definită ca disciplina prelucrării datelor cu ajutorul echipamentelor automate de prelucrare.

Principalele probleme care pot fi considerate ca aparţinînd informaticii sînt : culegerea datelor, pregătirea datelor, codificarea acestora, transmiterea lor, prelucrarea datelor pe echipamente, stocarea şi conservarea lor.

Problema dezvoltării explozive a informaticii şi a rolului ei în economie, administraţie, cercetare spaţială, strategie militară, ştiinţă, învăţămînt etc. este bine cunoscută şi de nespecialişti. Vom arăta numai că de la cîteva

---

* Sistemele deterministe au o comportare previzibilă, în timp ce sistemele probabilistice au o comportare aleatoare.

** Apariţia şi dezvoltarea ciberneticii (începînd din deceniul al 5-lea al secolului nostru) este legată de numele unor savanţi celebri ca Norbert Wiener, Claude Shannon, Ross Ashby etc.

*** Printre economiştii cibernetician din ţările socialiste, cunoscuţi pe plan mondial, putem cita de exemplu pe L. V. Kantarovici şi V.V. Novojilov (U.R.S.S.), O. Lange şi A. Greniewski (Polonia), Kornai Janos (Ungaria) etc.

Scanned with OKEN Scanner

calculatoare electronice și puțini specialiști în informatică, în 1945, s-a ajuns azi, pe plan mondial, la sute de mii de calculatoare și la peste un milion de specialiști.

● **Psihosociologia organizării** a apărut ca o nouă orientare în disciplinele conducerii în jurul anului 1950.

St. March, F. Simon și alți reprezentanți ai așa-numitei „școli psihosociologice" abordează, în principal, problema influenței factorilor psihologici și sociologici în compartimentul decizional. Luarea deciziilor, în concepția acestei școli, nu este funcție numai de criterii raționale ci și de modul de percepere a stimulilor, depinzînd de poziția decidentului și de relațiile cu ceilalți membri ai grupului.

Cu alte cuvinte, oricît s-ar face apel, în organizarea și conducerea organismelor economice, la metode și echipamente de mare finețe și tehnicitate, în ultimă instanță, oamenii sînt cei de care depinde funcționarea eficientă a sistemului, de aceea, trebuie studiate reacțiile individuale și relațiile dintre indivizii din sistem.

● **Teoria generală a sistemelor** (TGS), strîns legată de cibernetică, propune o perspectivă care să sintetizeze ideile viabile ale diferitelor orientări în științele organizării și conducerii.

Iată cîteva din ideile de bază ale teoriei generale a sistemelor, după „Industrial Dynamics" a lui J. Forrester:

a) orice sistem este alcătuit din elemente (părți) interdependente, acționînd în comun în virtutea unui scop;

b) ansamblul legăturilor între elementele sistemului, precum și al legăturilor cu întregul formează structura sistemului;

c) complexitatea sistemelor depinde mai mult de structura sistemului decît de natura părților sale;

d) două sisteme cu structuri parțial identice se numesc homomorfe (sistemul mai simplu va constitui un model al sistemului homomorf mai complex);

e) două sisteme homomorfe vor avea un comportament asemănător, de unde rezultă posibilitatea de studiu a proprietăților sistemelor reale prin simulare;

f) structura (statica) unui sistem preexistă comportamentului său (dinamicii sistemului);

g) mișcările într-un sistem se realizează prin fluxuri presupuse concrete și continue;

h) într-un organism economic toate categoriile de mișcări pot fi grupate în următoarele tipuri de fluxuri interconectate; 1) fluxuri materiale; 2) fluxuri de comenzi; 3) fluxuri bănești; 4) fluxuri umane; 5) fluxuri de echipamente și 6) fluxuri informaționale;

i) fluxul informațional are un rol central în funcționarea sistemelor;

j) procesele decizionale sînt considerate și ele ca avînd un rol central în mecanismul sistemelor; ele sînt presupuse a fi discontinue;

k) reglarea este un element caracteristic al funcționării sistemelor;

l) procesele care au loc în sistemele economice sînt, de regulă, neliniare.

Pe baza acestor premise, Forrester construiește un procedeu de descriere a comportamentului unei întreprinderi, care utilizează metode cibernetice, informatice, psihosociologice, precum și procedee de modelare

Scanned with OKEN Scanner

matematică. De asemenea, sînt folosite analogii fizice şi tehnice (de exemplu, fluxurile sînt examinate în sens hidraulic) iar simularea este utilizată ca un procedeu de bază în descrierea comportamentului sistemelor.

În linii mari în „Industrial Dynamics" se urmăreşte înţelegerea stării unui sistem cu ajutorul unor ecuaţii care descriu în timp intrările, transformările şi ieşirile din sistem, pentru cele şase tipuri de fluxuri amintite mai sus. (E vorba deci de găsirea funcţiilor de reacţie ale sistemului.) Pe baza acestei descrieri matematice se pot face simulări pe calculator, cu ajutorul cărora se prevede evoluţia sistemului.

Ideile şi procedeele TGS, impresionante prin complexitatea lor, sînt în curs de sedimentare metodologică şi experimentare practică.

Marea majoritate a propoziţiilor enumerate mai sus şi care stau la baza teoriei lui Forrester se regăsesc explicit sau implicit şi la baza metodologiilor practice de **analiză sistemică.** Conceptele de flux informaţional şi proces decizional sînt dominante şi în analiza sistemică la fel ca în TGS, iar urmărirea mecanismului transformării intrărilor în ieşiri constituie obiectul principal al analizei de sistem la fel ca şi al TGS. Procedeul folosit de analiza sistemică nu mai este însă matematic, ci bazat pe descrierea explicită, calitativă, a proceselor informaţional-decizionale. În plus, în practica analizei de sistem, odată cu proiectarea proceselor informaţionale şi în special a celor decizionale, se urmăreşte îmbunătăţirea lor, deci, se au în vedere criterii de optim. În acţiunea aceasta de proiectare eficientă a procesului informaţional-decizional, analiza sistemică face din plin apel la procedeele cercetării operaţionale şi la tehnicile informaticii.

În ceea ce priveşte folosirea metodelor psihosociologice, în analiza de sistem există încercări recente în acest sens.

1.2. Rolul modelării în cercetarea operaţională

Conceptul de „model", atît de mult folosit în ştiinţa modernă este relativ nou, dar metoda modelării este tot atît de veche pe cît sînt preocupările oamenilor pentru cunoaşterea ştiinţifică*.

Putem considera că modelul este o reprezentare izomorfă a realităţii, care oferind o imagine intuitivă şi totuşi riguroasă, în sensul structurii logice, a fenomenului studiat, facilitează descoperirea unor legături şi legităţi imposibil sau foarte greu de găsit pe alte căi.

În elaborarea modelelor economico-matematice, teoria economică are un rol deosebit de important întrucît ea formulează categoriile, conceptele şi legile obiective ale realităţii economice. Numai sprijinindu-se pe teoria economică modelele matematice pot reprezenta fidel fenomenele economice.

---

* Oamenii de ştiinţă din toate timpurile au folosit „modele" în cele mai diverse domenii ale cunoaşterii ştiinţifice. Pînă de curînd însă, la fel ca eroul lui Molière care „făcea proză fără să o ştie", ei utilizau modelarea fără a folosi termenul respectiv.

Scanned with OKEN Scanner

Modelul, ca instrument al cunoașterii științifice, este folosit în foarte numeroase discipline teoretice și practice. Fără pretenția de a face o clasificare riguroasă a tipurilor de modele, vom arăta că ele pot fi: modele verbal-descriptive — folosite în toate disciplinele nematematizate, modele matematice, modele fizice analogice (de tipul machetelor statice sau dinamice), modele grafice etc.

În științele economice, în special în disciplinele organizării și conducerii, modelele sînt utilizate în toată diversitatea de tipuri care există. În ultimele decenii însă, se conturează din ce în ce mai mult tendința utilizării cu precădere, în aceste discipline, a modelelor de tip matematic, datorită în special capacității acestora de a condensa riguros esențialul, cît și posibilității lor de a fi programate cu ajutorul calculatoarelor electronice, alcătuind împreună un instrument de investigație științifică de o putere necunoscută pînă în prezent, o prodigioasă „prelungire" a inteligenței umane.

O sistematizare metodologică a modelelor matematice întrebuințate în disciplinele organizării și conducerii social-economice ar fi riscantă avînd în vedere mutațiile continue și spectaculoase care au loc în aceste discipline și, în plus, ar avea un caracter pur scolastic, fără utilitate teoretică sau practică reală. De aceea, ne vom limita, în continuare, să enumerăm principalele tipuri de modele matematice cunoscute în acest domeniu.

După întinderea domeniului studiat, modelele care descriu realitatea economică pot fi: macroeconomice — cele care se referă la economia națională, la ramură (subramură) sau la economia unui teritoriu mare (un județ, o anumită zonă industrială, agricolă etc.) și microeconomice — la nivel de întreprindere, uzină, trust, combinat etc.

Modelele cibernetico-economice urmăresc să studieze relația dintre intrări și ieșiri într-un organism economic, cu evidențierea fenomenelor de reglare care determină buna funcționare a sistemului. Majoritatea modelelor cibernetico-economice sînt macroeconomice.

Modelele econometrice descriu comportamentul organismelor economice cu ajutorul unor sisteme de ecuații în care elementele numerice sînt determinate statistic. Și aceste modele sînt, de obicei, macroeconomice.

Modelele de simulare încearcă să stabilească modul de funcționare al unor organisme macro sau microeconomice prin acordarea unor combinații de valori întîmplătoare variabilelor independente care descriu procesele. Prin „citirea" valorilor pe care le capătă în felul acesta variabilele dependente, se obțin mărimi semnificative în procesul studiat.

Modelele sistemice au drept obiectiv surprinderea ansamblului aspectelor dintr-un organism economic (de exemplu, în modelele Forrester se consideră că prin identificarea celor șase fluxuri caracteristice — vezi cap. I — se poate cunoaște comportarea sistemului ca un tot).

Modelele cercetării operaționale se caracterizează prin căutarea unei soluții optime sau apropiată de optim, pentru fenomenul studiat. Modelele cercetării operaționale se bazează pe o mare diversitate de procedee matematice și au aplicații la nivel macro dar în special la nivel microeconomic. Ele reprezintă principalul instrument pentru optimizarea deciziilor în analiza de sistem.

Tipologia de mai sus este foarte relativă, între grupele menționate existînd frecvente asemănări și întrepătrunderi. Astfel, modelele econome-

Scanned with OKEN Scanner

trice sînt adesea de tip cibernetic, simularea se utilizează în mai toate tipurile de modele matematice, modelele cercetării operaţionale pot fi folosite în descrierea sistemică a unui organism etc.

Să examinăm, în continuare, procedeele practice de elaborare şi utilizare a modelelor matematice în disciplinele organizării şi conducerii.

În primul rînd trebuie subliniat faptul că activitatea de modelare, pentru a fi eficientă, trebuie desfăşurată întotdeauna în cadrul analizei de sistem şi anume ca un moment al etapei de proiectare a noului sistem. O serie de operaţii care se desfăşoară în cadrul analizei de sistem înaintea acestui moment au un caracter pregătitor pentru efectuarea modelării, iar altele, ulterioare ei, sînt necesare pentru aplicarea în practică a modelelor elaborate.

Să punctăm care sînt principalele faze ale elaborării unui model matematic într-o problemă de organizare-conducere social-economică, avînd grijă să evidenţiem cum se împletesc aceste faze cu alte operaţii ale analizei de sistem.

• Prima fază a modelării, care are un caracter pregătitor, este cunoaşterea realităţii în organismul studiat, în scopul îmbunătăţirii mecanismului informaţional-decizional. Descrierea logicii proceselor decizionale, alături de considerarea obiectivelor viitorului sistem, sînt principalele elemente ale cunoaşterii realităţii necesare modelării.

• A doua fază a modelării o reprezintă construirea propriu-zisă a modelului. Această operaţie, în marea majoritate a cazurilor din practică, constă în aplicarea unui instrument clasic de modelare ales din gama extrem de variată pe care ne-o pune la dispoziţie teoria cercetării operaţionale. În astfel de situaţii, abilitatea analistului constă în stabilirea corespondenţei dintre realitate şi instrumentul de modelare cunoscut din literatura de specialitate. Există şi cazuri cînd nu se poate stabili o astfel de corespondenţă, analistul fiind obligat să elaboreze modele noi. Acestea pot fi de două feluri : a) combinaţii de modele clasice, din domeniul teoriei şi b) modele noi propriu-zise. În primul caz, totul se reduce la buna cunoaştere a realităţii şi a teoriei, la care trebuie adăugată o doză de abilitate în combinarea metodelor. În cazul al doilea, este vorba despre creaţie originală. Elaborarea unui model matematic realmente original reclamă, pe lîngă profunda cunoaştere a realităţii care urmează a fi modelată, o foarte solidă cultură matematică, imaginaţie şi talent. După cum va rezulta din parcurgerea în prezentul curs a modelelor clasice ale cercetării operaţionale, există o mare diversitate în structura matematică şi logică a modelelor, de la modele foarte simple, neaxiomatizate, cum sînt cele din programarea liniară, la modele combinatorice în probleme de teoria grafelor, analiză a drumului critic şi programarea operativă a producţiei şi pînă la modele de mare fineţe, prezentate axiomatizat, cum sînt cele ale utilităţii sau deciziilor de grup. Evident elaborarea în formă axiomatizată a unui model reprezintă un stadiu superior în procesul modelării care însă nu poate fi totdeauna atins în practică.

Să examinăm pe scurt în ce constă modelarea în forma axiomatizată.

Un model axiomatizat (sistem axiomatic) cuprinde :

— axiomele sistemului, reprezentînd propoziţii exprimate în formă matematică, de regulă foarte puţine, care conţin unele adevăruri de mare

Scanned with OKEN Scanner

generalitate privind fenomenul care se modelează, atît de generale, încît toate constatările concrete şi particulare vor putea fi deduse din cele generale;

— reguli de inferenţă, reprezentînd prescripţii riguroase, singurele admise în sistem, prin care se trece de la axiome la teoreme, sau de la teoreme deja demonstrate, la altele noi;

— teoreme, adică propoziţii mai mult sau mai puţin particulare, exprimate matematic, deduse prin reguli de inferenţă din aproape în aproape din axiome, şi care exprimă proprietăţi ale fenomenului modelat.

Cînd în procesul de modelare axiomatică se explicitează limitativ conceptele care urmează a fi utilizate, adică se dă de la început o listă a noţiunilor şi operaţiilor matematice admise în sistem, se obţine o formă superioară de sistem axiomatic, numit *sistem formal*.

Sistemele formale sînt încă foarte puţin utilizate în ştiinţă şi cu atît mai puţin în disciplinele organizării şi conducerii economice.

Axiomatizarea şi, în ultima analiză, formalizarea, reprezintă viitorul în modelarea matematică, datorită rigorii excepţionale pe care o introduc, diminuării considerabile a elementelor de intuiţie şi arbitrar, care deşi mult mai puţine decît în modelele nematematizate, sînt încă prezente în modelarea matematică axiomatizată.

• A treia fază a modelării este confruntarea modelului cu realitatea şi eventual experimentarea sa. Această fază se realizează în cadrul implementării sistemului, care poate fi considerată a patra şi ultima fază a modelării.

În încheierea acestui paragraf, să examinăm cîteva caracteristici ale instrumentelor de modelare matematică pe care ni le pune la dispoziţie cercetarea operaţională.

Una din principalele caracteristici ale tuturor metodelor cercetării operaţionale este faptul că unele probleme ale cercetării operaţionale pot fi privite din perspectiva pur teoretică ca probleme de matematică pură. Evident, nu aceasta va fi perspectiva pe care o vom adopta în cele ce urmează întrucît vom privi metodele cercetării operaţionale strîns legate de problemele practice.

Din punct de vedere istoric, unele dintre problemele cercetării operaţionale s-au ivit, ce e drept, în special sub aspect pur matematic, cu mult înainte de a fi apărut activitatea organizată şi denumirea de cercetare operaţională. Astfel, unele noţiuni de teoria grafelor se cunosc de mai bine de un secol, teoria aşteptării îşi are originea în unele lucrări ale lui Erlang din deceniul al 2-lea al secolului nostru, iar teoria stocurilor apare către anul 1930. Ca disciplină de-sine-stătătoare, cercetarea operaţională a apărut însă în timpul celui de-al doilea război mondial, prin înfiinţarea unor echipe complexe (matematicieni, ingineri, economişti, biologi, psihologi, ş.a.) însărcinate cu optimizarea deciziilor privind unele acţiuni pregătitoare operaţiilor militare. După război, echipele astfel formate s-au reprofilat rapid pentru activităţi paşnice. Dezvoltîndu-se spectaculos în ultimele trei decenii, preocupările teoretice şi în special practice în domeniul cercetării operaţionale au ajuns să antreneze astăzi pe plan mondial sute de mii de specialişti.

În prezent nu se mai poate concepe conducerea unei activităţi tehnico-economice importante fără a face apel la metodele cercetării operaţionale,

Scanned with OKEN Scanner

bineînțeles împreună cu celelalte tehnici moderne cum ar fi informatica analiza de sistem ș.a.

În țara noastră, numeroși cercetători, cadre didactice din învățămîntul superior, specialiști din întreprinderi și organizații centrale și-au adus contribuția în aplicarea cercetării operaționale precum și în dezvoltarea originală a unora dintre capitolele acestei discipline complexe.

Programul Partidului Comunist Român de făurire a societății socialiste multilateral dezvoltate și înaintare a țării noastre spre comunism, precum și Programul—directivă de cercetare științifică, dezvoltare tehnologică și de introducere a progresului tehnic în perioada 1981 — 1990 și direcțiile principale pînă în anul 2000, prevăd o gamă largă de acțiuni pentru promovarea metodelor moderne de conducere, pentru aplicarea eficientă a cercetării științifice în domeniul optimizării deciziilor în economie. Există o gamă largă de probleme economice care se pretează a fi modelate matematic cu ajutorul procedeelor cercetării operaționale, dintre care menționăm: fundamentarea și optimizarea planului la diverse niveluri, programarea operativă a producției, repartizarea rațională a resurselor, eșalonarea în timp a activităților unei investiții, gestiunea stocurilor etc.

În contextul celor arătate trebuie subliniată importanța metodelor cercetării operaționale în fundamentarea și elaborarea planului cît și în urmărirea realizării acestuia.

Toate aceste probleme precum și numeroase altele au făcut obiectul aplicațiilor și exemplelor pe care le cuprinde cursul nostru. Spațiul editorial limitat a obligat pe autori să selecteze din vastul material pe care îl cuprinde cercetarea operațională azi, numai problemele de primă importanță teoretică, practică și didactică. Bibliografia recomandată permite cititorilor lărgirea orizontului și eventual, ca un prim pas pe linia cercetării științifice, unele incursiuni în profunzimea problemelor.

Scanned with OKEN Scanner

Capitolul II

PROGRAMAREA LINIARĂ

Printre metodele matematice larg utilizate în economie, un rol important îl are programarea matematică (programarea liniară, neliniară, parametrică, dinamică, stocastică etc.).

În general, scopul principal pe care-l urmăreşte programarea matematică constă în obţinerea soluţiei optime a unei probleme economice pe baza unui model matematic ale cărui caracteristici le vom prezenta în acest capitol.

Sfera de aplicare în economie a programării matematice cuprinde un număr mare şi variat de probleme privind planificarea economică operativă şi de perspectivă, conducerea diferitelor procese de producţie, dirijarea diferitelor procese tehnologice etc. Dintre toate ramurile programării matematice, programarea liniară este metoda cea mai răspîndită de rezolvare a unor probleme economice, pe de o parte, datorită caracterului relativ simplu al aparatului matematic folosit în modelare şi rezolvare, iar pe de altă parte, datorită faptului că este uşor accesibilă pentru reprezentarea matematică şi analiza fenomenelor economice.

2.1. Consideraţii generale privind modelul matematic al problemei de programare liniară

Modelul matematic de programare liniară utilizat în reprezentarea unui sistem economic este constituit dintr-un ansamblu de relaţii liniare, dintre care una reflectă obiectivul urmărit iar celelalte cuprind restricţiile economice sau tehnologice.

Pentru a înţelege mai bine modelul matematic al unei probleme de programare liniară, sub forma generală, vom analiza mai întîi un exemplu simplu.

O întreprindere dispune de două resurse R_1 şi R_2 în cantităţile $b_1 = 40$ unităţi şi respectiv $b_2 = 60$ unităţi. Din aceste resurse se fabrică două produse P_1 şi P_2 care aduc întreprinderii beneficiile unitare $c_1 = 10$ unităţi monetare şi respectiv $c_2 = 15$ unităţi monetare.

Scanned with OKEN Scanner

Coeficienţii de consum specific sînt daţi în tabelul 2.1 :

Tabelul 2.1

Resurse \ Produse	P_1	P_2
R_1	2	4
R_2	6	2

Întreprinderea trebuie să stabilească ce cantităţi va fabrica din fiecare produs în aşa fel încît beneficiul realizat să fie maxim.

Pentru a stabili modelul matematic al problemei formulate, notăm cu x_1 numărul de unităţi din produsul P_1 şi cu x_2 numărul de unităţi din produsul P_2 care vor fi fabricate pe baza resurselor existente. Beneficiul total, corespunzător nivelelor x_1 şi x_2 ale producţiei, se exprimă prin funcţia :

$$\mathbf{Z}(x_1, x_2) = 10x_1 + 15x_2.$$

Cum pentru o unitate din produsul P_1 se consumă 2 unităţi din resursa R_1 iar pentru o unitate din produsul P_2, 4 unităţi, va trebui să introducem următoarea restricţie impusă de limitarea resursei R_1 :

$$2x_1 + 4x_2 \leqslant 40.$$

Raţionînd analog pentru resursa R_2 obţinem restricţia :

$$6x_1 + 2x_2 \leqslant 60$$

iar din natura economică rezultă că $x_1 \geqslant 0$, $x_2 \geqslant 0$ (nu are sens fabricarea unei cantităţi negative).

Aşadar, modelul matematic al problemei considerate este :

$$\text{maxim } [\mathbf{Z}(x_1, x_2)] = 10x_1 + 15x_2^*$$

$$2x_1 + 4x_2 \leqslant 40$$

$$6x_1 + 2x_2 \leqslant 60$$

$$x_1 \geqslant 0, \quad x_2 \geqslant 0.$$

Remarcăm imediat faptul că în modelul matematic al problemei de mai sus, atît funcţia obiectiv cît şi restricţiile sînt funcţii liniare de necunoscutele x_1 şi x_2.

Considerăm că este util pentru cititor, ca înainte de a analiza diversele forme sub care se prezintă o problemă de programare liniară, să construim modelul corespunzător unor probleme economice concrete.

Exemplul 2.1. Repartizarea sarcinilor de producţie între întreprinderi în cazul unui singur produs.

---

* În continuare în cuprinsul lucrării se consideră că operatorul maxim sau minim este aplicat şi formei explicitate a funcţiei obiectiv (scrisă în membrul drept).

Scanned with OKEN Scanner

Un număr de m întreprinderi care fabrică acelaşi produs trebuie să aprovizioneze n beneficiari. În tabelul 2.2 sînt trecute cantităţile a_i $(1 \leqslant i \leqslant m)$ realizate de cele m întreprinderi, cantităţile b_j $(1 \leqslant j \leqslant n)$ de care au nevoie cei n beneficiari, costul transportului unei unităţi de produs de la întreprinderea A_i, $(1 \leqslant i \leqslant m)$ la beneficiarul B_j, $(1 \leqslant j \leqslant n)$ pe care l-am notat cu c_{ij}, precum şi costul de producţie p_i al unei unităţi din produsul întreprinderii A_i.

Tabelul 2.2

Întreprindere \ Beneficiar	B_1	B_2 . . .	B_j . . .	B_n	Disponibil	Cost unitar
A_1	c_{11}	c_{12} . . .	c_{1j} . . .	c_{1n}	a_1	p_1
A_2	c_{21}	c_{22} . . .	c_{2j} . . .	c_{2n}	a_2	p_2
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
A_i	c_{i1}	c_{i2}	c_{ij}	c_{in}	a_i	p_i
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
A_m	c_{m1}	c_{m2} . . .	c_{mj} . . .	c_{mn}	a_m	p_m
Necesar	b_1	b_2 . . .	b_j . . .	b_n	—	

Să se determine repartiţia optimă a sarcinilor de producţie pe întreprinderi astfel încît cheltuielile de transport şi costul de producţie pentru întreaga cantitate fabricată şi transportată să fie minim.

Dacă notăm cu x_{ij} cantitatea din producţia întreprinderii A_i care urmează a fi livrată beneficiarului B_j $(1 \leqslant i \leqslant m,\ 1 \leqslant j \leqslant n)$ atunci relaţiile:

$$x_{11} + x_{12} + \ldots + x_{1n} = a_1$$
$$x_{21} + x_{22} + \ldots + x_{2n} = a_2$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$x_{m1} + x_{m2} + \ldots + x_{mn} = a_m$$

exprimă faptul că suma cantităţilor livrate celor n beneficiari de către producătorul A_i trebuie să fie egală cu cantitatea produsă la această întreprindere $(1 \leqslant i \leqslant m)$.

Ţinînd acum seama de faptul că fiecare beneficiar trebuie să primească de la cei m producători o cantitate egală cu necesarul său, înseamnă că trebuie să fie satisfăcute şi relaţiile:

$$x_{11} + x_{21} + \ldots + x_{m1} = b_1$$
$$x_{12} + x_{22} + \ldots + x_{m2} = b_2$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$x_{1n} + x_{2n} + \ldots + x_{mn} = b_n.$$

Natura problemei impune evident că toate cantităţile x_{ij} sînt nenegative :

$$x_{ij} \geqslant 0 \quad (1 \leqslant i \leqslant m;\ 1 \leqslant j \leqslant n).$$

Problema care trebuie rezolvată cere să se determine cantităţile x_{ij} care satisfac condiţiile scrise mai sus astfel încît cheltuielile totale să fie minime, adică funcţia :

$$\mathbf{Z}(x_{11},\ x_{12},\ldots,x_{1n},\ x_{21},\ldots,x_{2n},\ldots,x_{m1},x_{m2},\ldots,x_{mn}) =$$
$$= (p_1 + c_{11})\,x_{11} + (p_1 + c_{12})\,x_{12} + \ldots + (p_m + c_{m1})\,x_{m1} + \ldots +$$
$$+ (p_m + c_{mn})\,x_{mn} \text{ să ia valoarea minimă.}$$

Aşadar, modelul matematic este format din :

1) funcţia de minimizat

$$\mathbf{Z}(x_{11}, x_{12}, \ldots, x_{m1}, \ldots, x_{mn}) = \sum_{i=1}^{m} \sum_{j=1}^{n} (p_i + c_{ij})\,x_{ij}$$

2) restricţiile :

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad 1 \leqslant i \leqslant m \qquad (m \text{ egalităţi})$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad 1 \leqslant j \leqslant n \qquad (n \text{ egalităţi})$$

3) condiţia de nenegativitate a variabilelor

$$x_{ij} \geqslant 0; \qquad 1 \leqslant i \leqslant m;\ 1 \leqslant j \leqslant n \qquad (m.n \text{ variabile}).$$

Exemplul 2.2. O cooperativă agricolă de producţie dispune de a_i tone îngrăşăminte de tipul $\mathbf{I}_i$ $(1 \leqslant i \leqslant m)$, de b unităţi forţă de muncă şi c hectare de pămînt.

Tabelul 2.3

Resurse \ Tipuri de legume	$\mathbf{L}_1$	$\mathbf{L}_2 \ldots$	$\mathbf{L}_j \ldots \mathbf{L}_n$
$\mathbf{I}_1$	a_{11}	$a_{12}\ldots$	$a_{1j}\ldots a_{1n}$
$\mathbf{I}_2$	a_{21}	$a_{22}\ldots$	$a_{2j}\ldots a_{2n}$
.	.	.	. .
.	.	.	. .
.	.	.	. .
$\mathbf{I}_i$	a_{i1}	$a_{i2}\ldots$	$a_{ij}\ldots a_{in}$
.	.	.	. .
.	.	.	. .
.	.	.	. .
$\mathbf{I}_m$	a_{m1}	$a_{m2}\ldots$	$a_{mj}\ldots a_{mn}$
Forţa de muncă	b_1	$b_2\ldots$	$b_j\ldots b_n$
Costul de producţie la hectar	p_1	$p_2\ldots$	$p_j\ldots p_n$

Scanned with OKEN Scanner

Sectorul legumicol al acestei C.A.P., în care se pot cultiva n tipuri de legume, dispune de cel mult a_i tone îngrăşăminte de tipul I_i, cel puţin $\bar{b}$ unităţi de forţă de muncă (indicator ce exprimă consumul de forţă de muncă în anul de bază) şi cel puţin $\bar{c}$ hectare de pămînt. În tabelul 2.3 sînt trecute consumurile de îngrăşăminte, forţă de muncă şi costul de producţie la hectar p_j pentru fiecare tip de legume j.

Se cere să se stabilească structura optimă a culturilor de legume astfel încît costul total al producţiei să fie minim.

Pe baza datelor problemei considerate alcătuim acum modelul matematic corespunzător.

Notăm cu x_j suprafaţa (exprimată în hectare) cultivată cu legume de tipul L_j $(1 \leqslant j \leqslant n)$.

Dacă a_{ij} reprezintă numărul de unităţi din îngrăşămîntul I_i necesar unui hectar din cultura L_j atunci avem restricţiile:

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\leqslant a_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\leqslant a_2 \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\leqslant a_m.
\end{aligned}$$

Forţa de muncă utilizată pentru cultivarea celor n tipuri de legume trebuie să fie cel puţin de $\bar{b}$ unităţi, dar în nici un caz nu trebuie să depăşească disponibilul de b unităţi al C.A.P. Aşadar trebuie satisfăcute relaţiile:

$$\begin{aligned}
b_1x_1 + b_2x_2 + \ldots + b_nx_n &\geqslant \bar{b} \\
b_1x_1 + b_2x_2 + \ldots + b_nx_n &\leqslant b.
\end{aligned}$$

Restricţiile referitoare la suprafaţa cultivată trebuie să ţină seama de suprafaţa planificată pentru legume care trebuie să fie de cel puţin $\bar{c}$ hectare, dar care nu trebuie însă să depăşească disponibilul C.A.P. de c ha.

$$\begin{aligned}
x_1 + x_2 + \ldots + x_n &\geqslant \bar{c} \\
x_1 + x_2 + \ldots + x_n &\leqslant c.
\end{aligned}$$

Cantităţile $x_1, x_2, \ldots, x_n \geqslant 0$ trebuie determinate astfel încît să fie satisfăcute restricţiile indicate şi în acelaşi timp să minimizeze costul total al producţiei, adică funcţia

$$Z(x_1, x_2, \ldots, x_n) = p_1x_1 + p_2x_2 + \ldots + p_nx_n.$$

Deci modelul matematic al problemei este format din:

1) funcţia obiectiv (ce trebuie minimizată)

$$\text{minim } [Z(x_1, x_2, \ldots, x_n)] = \sum_{j=1}^{n} p_j x_j$$

Scanned with OKEN Scanner

2) restricţiile problemei

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant a_i \qquad 1 \leqslant i \leqslant m$$

$$\underline{b} \leqslant \sum_{j=1}^{n} b_j x_j \leqslant \overline{b}$$

$$\underline{c} \leqslant \sum_{j=1}^{n} x_j \leqslant \overline{c}$$

3) condiţia de nenegativitate a variabilelor

$$x_j \geqslant 0, \quad 1 \leqslant j \leqslant n.$$

Exemplul 2.3. Într-un laborator al unei întreprinderi metalurgice se încearcă obţinerea unui aliaj special care trebuie să conţină cel puţin 4% Mn, cel puţin 3% Cr şi cel puţin 2% Ni. Întreprinderea dispune de două categorii de materii prime care conţin Mn, Cr şi Ni în cantităţile specificate în tabelul 2.4.

Costul unui kg de materie primă de tipul M_1 este 70 lei iar de tipul M_2 este de 55 lei.

Să se afle reţeta de amestec optimă din punctul de vedere al costului.

Tabelul 2.4

Materii prime \ Aliaje	Mn	Cr	Ni
M_1	2%	4%	2%
M_2	6%	3%	1%

Dacă notăm cu x_1 cantitatea din materia primă M_1 şi cu x_2 cantitatea de materie primă M_2 necesare amestecului atunci modelul matematic este:

$$\min [Z(x_1,x_2)] = 70x_1 + 55x_2$$
$$0{,}02x_1 + 0{,}06x_2 \geqslant 0{,}04(x_1 + x_2)$$
$$0{,}04x_1 + 0{,}03x_2 \geqslant 0{,}03(x_1 + x_2)$$
$$0{,}02x_1 + 0{,}01x_2 \geqslant 0{,}02(x_1 + x_2)$$
$$x_1, x_2 \geqslant 0.$$

Exemplul 2.4. O întreprindere fabrică patru categorii de produse P_1, P_2, P_3, P_4 utilizînd trei locuri de muncă L_1, L_2, L_3. Necesarul de timp mecanic exprimat în ore pentru fabricarea fiecărui produs, capacităţile

Scanned with OKEN Scanner

de producţie a celor trei locuri de muncă precum şi preţurile de desfacere a produselor sînt trecute în tabelul 2.5.

Tabelul 2.5

Locul de muncă \ Produsele	P_1	P_2	P_3	P_4	Capacităţile de producţie
L_1	45	20	20	9	3 780
L_2	45	25	25	0	4 410
L_3	0	5	0	0	350
Preţul de desfacere	1 260	1 960	700	1 000	—

Se cere să se determine structura optimă a producţiei.

Dacă notăm cu x_i numărul de produse P_i $(1 \leqslant i \leqslant 4)$, atunci modelul matematic este dat de :

$$\max \; [Z(x_1, x_2, x_3, x_4)] = 1\,260x_1 + 1\,960x_2 + 700x_3 + 1\,000x_4$$

$$45x_1 + 20x_2 + 20x_3 + 9x_4 \leqslant 3\,780$$

$$45x_1 + 25x_2 + 25x_3 \leqslant 4\,410$$

$$5x_2 \leqslant 350$$

$$x_1, x_2, x_3, x_4 \geqslant 0.$$

Caracterul de liniaritate al modelului este determinat de respectarea următoarelor ipoteze de natură economică şi matematică :

I_1) Cantităţile de resurse utilizate sînt proporţionale cu nivelul (intensitatea) activităţii ;

I_2) Fiecare resursă este caracterizată printr-o ecuaţie de echilibru, termenii acesteia reprezentînd fluxuri de intrări-ieşiri care se referă la activităţile ce consumă resursa respectivă.

I_3) Liniaritatea funcţiei obiectiv (funcţiei de eficienţă) care presupune proporţionalitate între nivelul fiecărei activităţi şi contribuţia sa la funcţia obiectiv.

I_4) Nenegativitatea nivelului activităţilor.

Pentru elaborarea modelului matematic trebuie să avem la dispoziţie o serie de informaţii care se materializează în constantele modelului. În vederea pregătirii informaţiilor necesare se parcurg o serie de etape dintre care amintim :

1. Analiza ansamblului de activităţi : descompunerea sistemului studiat în funcţii elementare de activităţi şi alegerea unităţilor de măsură a nivelului activităţii. Totodată se stabileşte caracteristica cu ajutorul căreia se apreciază eficienţa sistemului.

2. Determinarea categoriilor de mijloace de producţie $R_i (i = 1, 2, \ldots, m)$ care participă la fabricarea produsului* P_j $(j = 1, 2, \ldots, n)$ şi stabili-

* Produsul P_j este rezultatul activităţii a_j.

rea unităţilor de măsură a resurselor în concordanţă cu unitatea de măsură a nivelului activităţilor.

3. Determinarea coeficienţilor de consum specific a_{ij}, cunoscuţi şi sub denumirea de coeficienţi tehnologici. Aceştia constituie factorii de proporţionalitate între nivelurile x_j ale activităţilor şi fluxurile de resurse.

4. Determinarea relaţiilor dintre sistem considerat ca un tot şi exterior, în sensul stabilirii fluxurilor intrări-ieşiri.

5. Stabilirea relaţiilor de echilibru.

Formularea riguroasă din punct de vedere matematic a problemei de programare liniară se datoreşte lui G.B. Dantzig care a pus în evidenţă algoritmul de rezolvare cunoscut astăzi sub numele de algoritmul simplex.

2.2. Forme de prezentare a modelului matematic

Din analiza modelelor prezentate în paragraful precedent rezultă că într-o problemă de programare liniară pot apărea restricţii scrise sub formă de inegalităţi (de tipul „$\leqslant$" sau „$\geqslant$") sau egalităţi. De asemenea, criteriul de optimizare ales impune în unele cazuri maximizarea funcţiei obiectiv iar în alte cazuri, minimizarea acesteia.

Pentru studiul şi interpretarea soluţiilor unei probleme de programare liniară prezintă un deosebit interes formele canonice şi forma standard. În continuare vom prezenta aceste noţiuni.

Definiţia 1. O problemă de programare liniară este dată sub forma standard dacă toate restricţiile sînt ecuaţii (egalităţi) şi tuturor variabilelor li se impun condiţiile de nenegativitate.

Am definit dintr-un început forma standard a unei probleme de programare liniară întrucît orice problemă de programare liniară pentru a fi rezolvată cu ajutorul algoritmului simplex (sau simplex dual) va trebui adusă mai întîi la forma standard.

Astfel problema :

$$\begin{aligned} \max\ [\mathbf{Z}(x)] &= 3x_1 + 4x_2 \quad + x_3 \\ x_1 &\quad + x_4 \quad + x_5 = 6 \\ x_2 &\quad - 2x_4 \quad + x_5 = 0 \\ x_3 &\quad + x_4 \quad + 3x_5 = 2 \\ x_j \geqslant 0 &\qquad 1 \leqslant j \leqslant 5 \end{aligned}$$

este o problemă de programare liniară dată sub forma standard. Dacă introducem notaţiile :

$$\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & -2 & 1 \\ 0 & 0 & 1 & 1 & 3 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 6 \\ 0 \\ 2 \end{pmatrix}, \mathbf{C} = \begin{pmatrix} 3 \\ 4 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{X} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}$$

Scanned with OKEN Scanner

atunci problema considerată poate fi scrisă sub forma matriceală astfel :

$$\max\ [\mathbf{Z}(x)] = (3\ 4\ 1\ 0\ 0) \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & -2 & 1 \\ 0 & 0 & 1 & 1 & 3 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 0 \\ 2 \end{pmatrix}; \quad \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} \geqslant 0.$$

Prin generalizare, urmează că modelul matematic al unei probleme de programare liniară dată sub forma standard este următorul :

$$\min\ (\max)\ [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$

$$\mathbf{AX} = \mathbf{b}$$

$$\mathbf{X} \geqslant 0$$

unde **A** este o matrice cu m linii şi n coloane pe care o vom numi matricea coeficienţilor tehnologici, **X** este un vector coloană cu n componente — vectorul activităţilor, **b** un vector coloană cu m componente pe care-l vom numi de acum încolo vectorul „resurselor" iar **C** un vector coloană cu n componente — vectorul „costurilor". Prin **C'X** am notat produsul scalar al vectorilor **C'** şi **X**.

$$\mathbf{C}'\mathbf{X} = \sum_{j=1}^{n} c_j x_j$$

$$\mathbf{C}' = (c_1,\ c_2,\ \ldots,\ c_n);\quad \mathbf{X} = (x_1,\ x_2, \ldots, x_n).$$

Definiţia 2. Spunem că o problemă de programare liniară este dată sub forma canonică dacă toate restricţiile sînt inegalităţi de acelaşi sens şi tuturor variabilelor li se impun condiţii de nenegativitate ; dacă sensul restricţiilor este „$\geqslant$", atunci funcţia obiectiv trebuie minimizată iar dacă sensul restricţiilor este de tipul „$\leqslant$" funcţia obiectiv trebuie maximizată.

În ambele cazuri, restricţiile acestor probleme se numesc concordante.

Dacă exprimăm definiţia de mai sus într-o formă matematică cît mai concisă obţinem forma I sau II a unei probleme de programare liniară după cum urmează :

$$\min\ [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X} \qquad \max\ [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$

$$\mathbf{AX} \geqslant \mathbf{b} \qquad \mathbf{AX} \leqslant \mathbf{b}$$

$$\text{(I)}\ \ \mathbf{X} \geqslant 0 \qquad \text{(II)}\ \ \mathbf{X} \geqslant 0.$$

Exemplele 2.3 şi 2.4 au condus la forme canonice de tipul I şi respectiv II.

Scanned with OKEN Scanner

Trebuie precizat că în aplicațiile practice apar frecvent situații în care modelul conține simultan restricții de toate tipurile. Restricțiile de forma „$\geqslant$" sînt neconcordante pentru problemele la care se cere maximizarea funcției obiectiv, iar cele de tipul „$\leqslant$" sînt neconcordante pentru problemele la care se cere minimizarea funcției obiectiv. Dar orice problemă de programare liniară se poate aduce la forma standard sau la forma canonică cu ajutorul unor transformări elementare efectuate asupra variabilelor sau (și) restricțiilor precum și a operatorului aplicat funcției obiectiv. Aceste transformări sînt :

a) Transformarea operatorului maxim în minim (sau invers) se bazează pe egalitatea

$$\max_{x \in \mathbf{X}} [\mathbf{Z}(x)] = -\min_{x \in \mathbf{X}} \{-\mathbf{Z}(x)\}$$

adevărată pentru orice mulțime $\mathbf{X}$ și orice funcție reală definită pe $\mathbf{X}$. În particular :

$$\max \mathbf{C}'\mathbf{X} = -\min (-\mathbf{C})'\mathbf{X}.$$

b) Transformarea sensului unei inegalități se realizează prin înmulțirea ei cu -1.

c) O variabilă x, căreia nu i se impune restricție de semn, se înlocuiește prin diferența a două variabile nenegative :

$$x = x_1 - x_2, \text{ cu } x_1 \geqslant 0,\ x_2 \geqslant 0.$$

d) Ecuațiile se transformă în inecuații pe baza faptului că $a'x = b$ este echivalentă cu următoarele inecuații de sens contrar*

$$a'x \geqslant b \text{ și } a'x \leqslant b.$$

e) Inecuațiile se transformă în ecuații ținînd seama de faptul că inecuația $a'x \leqslant b$ poate fi scrisă ca o ecuație $a'x + y = b$ introducînd variabile ecart (sau variabile de compensare) $y \geqslant 0$. În mod analog o inecuație de forma $a'x \geqslant b$ se transformă în ecuația $a'x - y = b$ prin scăderea variabilei ecart $y \geqslant 0$. Trebuie să menționăm faptul că variabilele ecart nu apar în funcția obiectiv, sau altfel spus ele apar în funcția obiectiv cu coeficienții $c_j^e = 0$ (c_j^e coeficienții corespunzători variabilelor ecart).

Transformările menționate mai sus ne dau posibilitatea să obținem forma standard sau canonică echivalentă oricărei probleme de programare liniară.

Importanța utilizării acestor transformări nu constă numai în stabilirea echivalenței despre care am amintit mai sus. Ele vor apărea într-un mod mai concret atunci cînd se va pune problema utilizării algoritmului simplex pentru rezolvarea unei probleme sau cînd se vor considera probleme duale ale problemelor de programare liniară.

---

* S-a utilizat notația vectorială $a'x$, unde a' este vector linie, transpusul lui a, iar x este vector coloană.

Scanned with OKEN Scanner

2.3. Teoreme fundamentale

Teorema 1. Dacă problema de programare liniară

$$\min \text{ (max) } [\mathbf{Z(X)}] = \mathbf{C'X}$$

$$\mathbf{AX} = \mathbf{b}$$

$$\mathbf{X} \geqslant 0$$

admite un program (soluție admisibilă), atunci ea are cel puțin un program (soluție admisibilă) de bază.

Demonstrație. Fie $\mathbf{X} = (x_1, x_2, \ldots, x_k, 0, \ldots, 0)'$ un program* al problemei de programare considerate unde am pus componentele pozitive, pe primele k locuri în vectorul $\mathbf{X}$, $k = 0, 1, 2, \ldots, n$ (aceasta nu restrînge cu nimic generalitatea).

Dacă $k = 0$ demonstrația este încheiată căci $\mathbf{X} = 0$ este prin definiție un program de bază.

Fie deci $k > 0$. Atunci sînt posibile două situații:

1.1. Vectorii $a_1, a_2, \ldots, a_k$ (vectorii coloană din matricea $\mathbf{A}$ corespunzători componentelor pozitive $x_1, x_2, \ldots, x_k$) sînt liniar independenți. În acest caz $\mathbf{X}$ este un program de bază.

Se poate acum preciza și baza $\mathbf{B}$ corespunzătoare acestui program. Dacă $k = m$ atunci este evidentă baza $\mathbf{B} = (a_1, a_2, \ldots, a_m)$. Dacă însă $k < m$ atunci deoarece s-a presupus rang $\mathbf{A} = m$, există vectorii $a_{i_{k+1}}, a_{i_{k+2}}, \ldots, a_{i_m}$ liniar independenți care împreună cu vectorii $a_1, a_2, \ldots, a_k$ formează o bază și prin urmare în acest caz baza asociată programului de bază considerat, nu este unică.

1.2. Vectorii $a_1, a_2, \ldots, a_k$ sînt liniar dependenți ceea ce înseamnă că există un vector $\mathbf{Y} \in \mathbf{R}^n$, $\mathbf{Y} = (y_1, y_2, \ldots, y_k, 0, \ldots, 0)$; $\mathbf{Y} \neq 0$, pentru care $\mathbf{AY} = 0$.

Putem presupune că cel puțin o componentă $y_i < 0$ (în caz contrar raționăm cu vectorul $\mathbf{Z} = -\mathbf{Y}$).

În acest caz putem scrie:

$$\mathbf{AX} + \lambda \mathbf{AY} = \mathbf{A}(\mathbf{X} + \lambda \mathbf{Y}) = b$$

și deci $\mathbf{X} + \lambda \mathbf{Y}$ va fi un program al problemei de programare liniară considerate, dacă $\lambda \geqslant 0$ este ales astfel încît

$$\mathbf{X} + \lambda \mathbf{Y} \geqslant 0.$$

În particular pentru

$$\lambda = \lambda_0 = \min_{i, y_i < 0} \left\{ \frac{x_i}{-y_i} \right\}$$

---

* Un vector $\mathbf{X}$ care satisface restricțiile $\mathbf{AX} = b$, fără a satisface $\mathbf{X} \geqslant 0$, se numește soluție neadmisibilă (nerealizabilă). Dacă $\mathbf{X}$ satisface și restricțiile $\mathbf{X} \geqslant 0$ atunci el se numește program sau soluție admisibilă (realizabilă), iar dacă în plus realizează optimul funcției $\mathbf{Z(X)}$, vectorul $\mathbf{X}$ se numește program optim sau soluție optimă. Presupunem în plus că $m < n$ adică numărul restricțiilor este mai mare decît numărul variabilelor.

Scanned with OKEN Scanner

obținem programul $\mathbf{X} + \lambda_0 \mathbf{Y}$ care are cel mult $k-1$ componente strict pozitive.

Coloanele corespunzătoare acestor componente sînt sau liniar independente și atunci am obținut un program de bază, sau sînt liniar dependente și atunci reluăm raționamentul făcut în acest caz.

După un număr finit de pași se ajunge la cazul (*1.1*) și deci se obține un program de bază. Cu aceasta teorema este demonstrată.

Teorema 2. Dacă problema de programare liniară

$$\min (\max) \ [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$

$$\mathbf{AX} = b$$

$$\mathbf{X} \geqslant 0$$

admite un program optim atunci are un program de bază optim.

Demonstrație: Fie $\mathbf{X} = (x_1, x_2, \ldots, x_k, 0, \ldots, 0)'$ un program optim (unde s-a presupus ca și pînă acum că primele k componente sînt strict pozitive).

Dacă $k = 0$ demonstrația s-a încheiat.

Dacă $k > 0$ sînt posibile două situații:

2.1) Vectorii $a_1, a_2, \ldots, a_k$ sînt liniar independenți și atunci $\mathbf{X}$ este un program de bază optim.

2.2) Vectorii $a_1, a_2, \ldots, a_k$ sînt liniar dependenți și atunci există $\mathbf{Y} = (y_1, y_2, \ldots, y_k, 0, \ldots, 0)' \in \mathbf{R}^n$ astfel încît

$$\mathbf{AY} = 0$$

și deci există programul $\mathbf{X} + \lambda \mathbf{Y}$ pentru λ convenabil ales, adică pentru $\lambda \in [\lambda_1, \lambda_2]$ unde

$$\lambda_1 = \begin{cases} \max\limits_{i/y_i > 0} \left\{ -\dfrac{x_i}{y_i} \right\} \\ -\infty \quad \text{dacă toți } y_i \leqslant 0 \end{cases}$$

$$\lambda_2 = \begin{cases} \min\limits_{i/y_i < 0} \left\{ \dfrac{x_i}{-y_i} \right\} \\ +\infty \text{ dacă toți } y_i \geqslant 0. \end{cases}$$

Din faptul că $\mathbf{X}$ este program optim și $\mathbf{X} + \lambda \mathbf{Y}$ este program, rezultă că $\mathbf{C}'\mathbf{Y} = 0$, deoarece în caz contrar alegînd pe λ de semn contrar valorii $\mathbf{C}'\mathbf{Y}$ vom avea

$$\mathbf{C}'(\mathbf{X} + \lambda \mathbf{Y}) = \mathbf{C}'\mathbf{X} + \lambda \mathbf{C}'\mathbf{Y} < \mathbf{C}'\mathbf{X}$$

ceea ce ar contrazice faptul că $\mathbf{X}$ este program optim. Prin urmare $\mathbf{C}'(\mathbf{X} + \lambda \mathbf{Y}) = \mathbf{C}'\mathbf{X}$ adică $\mathbf{X} + \lambda \mathbf{Y}$ este program optim.

Putem alege acum pe $\lambda_0 = \lambda_1$ sau $\lambda_0 = \lambda_2$ astfel încît programul optim să aibă cel mult $k - 1$ componente strict pozitive.

Scanned with OKEN Scanner

Cei $k-1$ vectori rămaşi pot fi liniar independenţi şi atunci teorema este demonstrată sau pot fi liniar dependenţi şi atunci se repetă raţionamentul făcut. După un număr finit de paşi se ajunge la cazul (*2.1*) ceea ce înseamnă că am obţinut un program de bază optim.

Teorema 3. Mulţimea soluţiilor admisibile ale unei probleme de programare liniară este o mulţime convexă.

Demonstraţie. Pentru a ne fixa ideile să presupunem că problema este dată sub forma standard. (Se procedează în mod analog dacă problema este dată sub forma canonică.)

$$(\min)\max[\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$

Mulţimea soluţiilor admisibile este

$$\{\mathbf{X} \in \mathbf{R}^n : \mathbf{AX} = \mathbf{b},\ \mathbf{X} \geqslant 0\}.$$

Fie $\mathbf{X}_1, \mathbf{X}_2$, două astfel de soluţii. Atunci pentru $\lambda \in [0,1]$, $\mathbf{X} = \lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2 \geqslant 0$ şi în plus

$$\mathbf{AX} = \mathbf{A}[\lambda\mathbf{X}_1 + (1-\lambda)\mathbf{X}_2] = \lambda\mathbf{AX}_1 + (1-\lambda)\mathbf{AX}_2 =$$

$$= \lambda\mathbf{b} + (1-\lambda)\mathbf{b} = \mathbf{b}$$

ceea ce dovedeşte afirmaţia.

Teorema 4. Funcţia obiectiv a unei probleme de programare liniară ia valoarea optimă într-un punct extremal al mulţimii convexe $\mathbf{D}$ a tuturor soluţiilor admisibile ale problemei.

Demonstraţie. Presupunem, pentru a ne fixa, că se cere minimizată funcţia obiectiv şi să notăm cu $\widetilde{\mathbf{D}} = \{\mathbf{X}_i,\ i \in \mathbf{I}\}$ mulţimea punctelor extremale ale lui $\mathbf{D}$. Fie $\mathbf{X}_1$ un punct extremal (sau unul dintre ele dacă sînt mai multe) pentru care

$$\mathbf{C}'\mathbf{X}_1 = \min_{i \in \mathbf{I}} \{\mathbf{C}'\mathbf{X}_i\} = \bar{Z}_1$$

Presupunem că nu ar fi adevărat, deci că ar exista

$$\mathbf{X}_0 \in \mathbf{D} - \widetilde{\mathbf{D}} \text{ astfel încît}$$

$$\mathbf{C}'\mathbf{X}_0 = \bar{Z}_0 < \bar{Z}_1.$$

Deoarece $\mathbf{X}_0$ nu este extremal, înseamnă că se poate exprima în modul următor :

$$\mathbf{X}_0 = \sum_{i \in \mathbf{I}} \lambda_i \mathbf{X}_i,\ \lambda_i \geqslant 0, \quad \sum_{i \in \mathbf{I}} \lambda_i = 1.$$

Însă

$$\bar{Z}_0 = \mathbf{C}'\mathbf{X}_0 = \sum_{i \in \mathbf{I}} \lambda_i \mathbf{C}'\mathbf{X}_i \geqslant \sum_{i \in \mathbf{I}} \lambda_i \mathbf{C}'\mathbf{X}_1 = \mathbf{C}'\mathbf{X}_1 = \bar{Z}_1$$

care constituie o contradicţie. Prin urmare $\mathbf{X}_1$ este un program minimal.

Scanned with OKEN Scanner

Consecinţă. Dacă funcţia obiectiv ia aceeaşi valoare optimă în mai multe puncte extremale, atunci orice combinaţie liniară convexă* a acestora este soluţie optimă a problemei de programare liniară.

Demonstraţie. Într-adevăr, fie $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k$, puncte extremale pentru care

$$\mathbf{C}'\mathbf{X}_1 = \mathbf{C}'\mathbf{X}_2 = \ldots = \mathbf{C}'\mathbf{X}_k = \min_{i \in \mathbf{I}} \{\mathbf{C}'\mathbf{X}_i\} = \bar{\mathbf{Z}}$$

şi fie combinaţia liniară convexă $\sum_{i=1}^{k} \lambda_i \mathbf{X}_i = \mathbf{X}$.

Am văzut că $\mathbf{X}$ este însă soluţie admisibilă iar funcţia obiectiv are valoarea

$$\mathbf{C}'\mathbf{X} = \sum_{i=1}^{k} \lambda_i \mathbf{C}'\mathbf{X}_i = \bar{\mathbf{Z}} \sum_{i=1}^{k} \lambda_i = \bar{\mathbf{Z}} \cdot$$

2.4. Interpretarea geometrică a problemelor de programare liniară

Pentru înţelegerea teoremelor prezentate în paragraful precedent şi a tehnicii de rezolvare a unei probleme de programare liniară, vom analiza interpretarea geometrică în spaţiul activităţilor ($\mathbf{R}^n$). În cazul cînd $n = 2$, datele problemei pot fi reprezentate grafic în plan. Modelul matematic ataşat problemei date în tabelul 2.1 are forma :

$$\max [\mathbf{Z}(x)] = 10x_1 + 15x_2$$

în condiţiile

$$2x_1 + 4x_2 \leqslant 40$$

$$6x_1 + 2x_2 \leqslant 60$$

$$x_1 \geqslant 0\,; \quad x_2 \geqslant 0.$$

Ecuaţia $2x_1 + 4x_2 = 40$ defineşte o dreaptă pe care o vom nota cu D_1. Această dreaptă împarte planul $x_1 0 x_2$ în două semiplane : $2x_1 + 4x_2 \leqslant 40$ şi $2x_1 + 4x_2 > 40$. Pentru aflarea soluţiei optime a problemei analizate interesează doar punctele din semiplanul $2x_1 + 4x_2 \leqslant 40$. În acelaşi mod se interpretează cea de a doua restricţie. Reprezentînd cele două drepte (D_1 şi D_2) într-un sistem de axe carteziene $x_1 o x_2$ şi luînd în consideraţie condiţiile de nenegativitate $x_1 \geqslant 0$ şi $x_2 \geqslant 0$ precum

---

* *Definiţie.* Spunem că $\mathbf{X}$ este o combinaţie liniară convexă de punctele extremale $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_k$ dacă există $\lambda_i \geqslant 0$, $1 \leqslant i \leqslant k$, $\sum_{i=1}^{k} \lambda_i = 1$ astfel încît $\mathbf{X} = \sum_{i=1}^{k} \lambda_i \mathbf{X}_i$.

Scanned with OKEN Scanner

și intersecția semiplanelor considerate se obține poliedrul convex din figura *2.1*.

Funcția obiectiv $10x_1+15x_2=$ $=\mathbf{Z}$ determină, pentru diferite valori ale lui **Z**, o mulțime de drepte paralele care intersectează poliedrul convex **D**. Se observă că maximul funcției obiectiv $\mathbf{Z} = 170$ se obține în punctul de coordonate $x_1 = 8$ și $x_2 = 6$. Se constată că optimul problemei de programare liniară este atins într-unul din vîrfurile mulțimii convexe, **D**. Dacă se modifică funcția obiectiv (deci panta dreptei $\mathbf{C'X} = \mathbf{Z}$) rezultă o altă soluție optimă.

Să luăm un alt exemplu, ceva mai complicat desigur, pentru a putea prinde și alte aspecte ale interpretării geometrice a unei probleme de programare liniară.

Pentru aceasta considerăm problema de programare liniară dată sub forma standard :

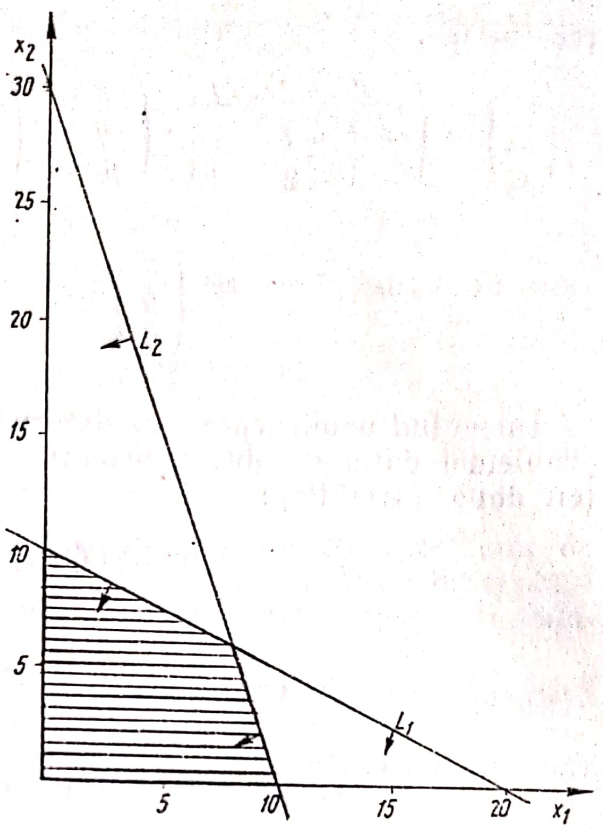


Fig. 2.1

$$\max(2x_1 + 3x_2 + 3x_3 - x_4 + 3x_5)$$
$$x_2 + x_3 + x_5 = 3$$
$$3x_1 + x_2 - 2x_3 + x_4 - 3x_5 = 2$$
$$5x_1 + 10\,x_2 + x_3 + 2x_4 = 20$$
$$x_j \geqslant 0 \qquad j = 1, 2, 3, 4, 5.$$

Deoarece $\mathbf{X} \in \mathbf{R}^5$, fiecare restricție va defini un hiperplan. Desigur este posibilă o interpretare geometrică în $\mathbf{R}^5$ unde mulțimea tuturor soluțiilor posibile este intersecția celor trei hiperplane (date în modelul nostru prin cele trei ecuații liniare) din care se ia numai partea cuprinsă în „octantul nenegativ" definit prin $x_j \geqslant 0$, $j = 1, 2, 3, 4, 5$.

În acest caz interpretarea se reduce la o problemă de limbaj, deoarece imaginea grafică atașată modelului este imposibil de realizat.

Totuși în acest caz este încă posibilă o interpretare geometrică în plan. Pentru aceasta vom alege pe x_1 și x_2 ca variabile secundare și prin urmare ca variabile de bază pe x_3, x_4, x_5, ceea ce este posibil după cum vom constata imediat.

Introducînd notațiile :

$$\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 \\ -2 & 1 & -3 \\ 1 & 2 & 0 \end{pmatrix}; \mathbf{X^B} = \begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix}; \mathbf{S} = \begin{pmatrix} 0 & 1 \\ 3 & 1 \\ 5 & 10 \end{pmatrix}; \mathbf{X^S} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}; \mathbf{b} = \begin{pmatrix} 3 \\ 2 \\ 20 \end{pmatrix}$$

Scanned with OKEN Scanner

rezultă că :

$$\begin{pmatrix} x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 6 & 2 & -1 \\ -3 & -1 & 1 \\ -5 & -2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 3 \\ 2 \\ 20 \end{pmatrix} - \begin{pmatrix} 6 & 2 & -1 \\ -3 & -1 & 1 \\ -5 & -2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 3 & 1 \\ 5 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} =$$

$$= \begin{pmatrix} 2 \\ 9 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 & -2 \\ 2 & 6 \\ -1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}.$$

Înlocuind acum x_3, x_4, x_5 determinaţi mai sus în funcţia obiectiv a problemei date, se obţine următoarea problemă de programare liniară (cu două variabile) :

$$\max\ [\mathbf{Z}(x)] = 4x_1 + 6x_2$$

în condiţiile

$$\begin{aligned} x_1 - 2x_2 &\leqslant 2 \\ 2x_1 + 6x_2 &\leqslant 9 \\ -x_1 + 3x_2 &\leqslant 1 \\ x_1 \geqslant 0,\ x_2 &\geqslant 0 \end{aligned}$$

care se poate reprezenta grafic în sistemul de axe carteziene $x_1 0 x_2$.

În acest mod se poate proceda cu orice problemă de programare liniară dată sub forma standard pentru care $n - m = 2$.

Aşadar, în cel mai general caz dacă avem o problemă de programare liniară dată sub forma standard

$$\max[\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$

$$\mathbf{AX} = \mathbf{b}$$

$$\mathbf{X} \geqslant 0$$

alegînd o bază oarecare **B** putem scrie :

$$\max[\mathbf{Z}(x)] = (\mathbf{C}^{\mathbf{B}})'\,\mathbf{X}^{\mathbf{B}} + (\mathbf{C}^{\mathbf{S}})'\,\mathbf{X}^{\mathbf{S}}$$

$$(\mathbf{B},\mathbf{S}) \begin{pmatrix} \mathbf{X}^{\mathbf{B}} \\ \mathbf{X}^{\mathbf{S}} \end{pmatrix} = \mathbf{b}$$

$$\begin{pmatrix} \mathbf{X}^{\mathbf{B}} \\ \mathbf{X}^{\mathbf{S}} \end{pmatrix} \geqslant 0$$

adică

$$\max[\mathbf{Z}(x)] = (\mathbf{C}^{\mathbf{B}})'\,\mathbf{X}^{\mathbf{B}} + (\mathbf{C}^{\mathbf{S}})'\mathbf{X}^{\mathbf{S}}$$

$$\mathbf{B}\mathbf{X}^{\mathbf{B}} + \mathbf{S}\mathbf{X}^{\mathbf{S}} = \mathbf{b}$$

$$\mathbf{X} \geqslant 0$$

Scanned with OKEN Scanner

sau încă, înmulțind la stînga sistemul de restricție cu $\mathbf{B}^{-1}$ și făcînd înlocuirile necesare, obținem

$$\max[\mathbf{Z}(x)] = (\mathbf{C}^{\mathbf{B}})'[\mathbf{B}^{-1}\,\mathbf{b} - \mathbf{B}^{-1}\mathbf{S}\mathbf{X}^{\mathbf{S}}] + (\mathbf{C}^{\mathbf{S}})'\,\mathbf{X}^{\mathbf{S}}$$

$$\mathbf{X}^{\mathbf{B}} = \mathbf{B}^{-1}\,\mathbf{b} - \mathbf{B}^{-1}\,\mathbf{S}\mathbf{X}^{\mathbf{S}}$$

$$\mathbf{X}^{\mathbf{B}} \geqslant 0,\quad \mathbf{X}^{\mathbf{S}} \geqslant 0$$

care ne conduce la următoarea problemă de programare liniară sub forma canonică, echivalentă cu cea dată :

$$\max[\mathbf{Z}(x)] = [(\mathbf{C}^{\mathbf{S}})' - (\mathbf{C}^{\mathbf{B}})'\mathbf{B}^{-1}\mathbf{S}]\;\mathbf{X}^{\mathbf{S}}$$

$$\mathbf{B}^{-1}\mathbf{S}\mathbf{X}^{\mathbf{S}} \leqslant \mathbf{B}^{-1}\,\mathbf{b}$$

$$\mathbf{X}^{\mathbf{S}} \geqslant 0.$$

De aici rezultă că mulțimea soluțiilor posibile ale unei probleme de programare liniară se poate interpreta fie ca o intersecție de hiperplane în spațiul activităților $\mathbf{R}^n$ fie ca un tronson în spațiul $\mathbf{R}^{n-m}$ (spațiul variabilelor secundare $\mathbf{X}^{\mathbf{S}}$).

Definiție. Se numește tronson, intersecția unui număr finit de semispații $a'x \leqslant b_i,\ i \in \mathbf{M} = \{1, 2, \ldots, m\}$.

Prin urmare tronsonul este mulțimea soluțiilor unui sistem de inecuații liniare. Un tronson mărginit se numește poliedru convex.

2.5. Algoritmul Simplex

După cum s-a arătat în paragrafele precedente, o posibilitate de rezolvare a problemelor de programare liniară constă în determinarea tuturor bazelor **B** din matricea **A**. Din mulțimea soluțiilor de bază (corespunzătoare mulțimii bazelor) reținem submulțimea soluțiilor pentru care $\mathbf{B}^{-1}\mathbf{b} \geqslant 0$, adică submulțimea soluțiilor posibile de bază, și dintre acestea alegem pe aceea sau pe acelea care minimizează (sau maximizează) funcția obiectiv. Deși acest procedeu ar putea fi aplicabil, din punctul de vedere practic el este cu totul ineficient deoarece chiar și în cazul unor probleme de programare liniară de dimensiuni mici este necesar să se investigheze un număr de $\mathbf{C}_n^m$ baze.

Uneori aceste calcule s-ar putea să fie inutile sau atunci cînd optimul este infinit, numai considerarea programelor de bază nu pune în evidență această situație.

În vederea înlăturării acestui neajuns se folosește o metodă de explorare sistematică și eficientă a programelor de bază, metodă care poartă numele de *algoritmul simplex*.

Mai precis algoritmul simplex ne dă posibilitatea să trecem succesiv de la un program de bază la altul, astfel încît valoarea funcției obiectiv să scadă, atunci cînd se minimizează funcția obiectiv (sau să crească cînd funcția obiectiv se maximizează).

Scanned with OKEN Scanner

În plus algoritmul simplex oferă criterii pe baza cărora putem decide dacă problema de programare liniară nu are soluţie optimă sau are optim infinit.

Pentru prezentarea algoritmului simplex vom utiliza forma standard a unei probleme de programare liniară, adică :

$$\max(\min)[\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$

$$\mathbf{AX} = \mathbf{b}$$

$$\mathbf{X} \geqslant 0.$$

Dacă $\mathbf{B}$ este o bază (vom presupune că $\mathbf{B}^{-1} \cdot \mathbf{b} \geqslant 0$) atunci sistemul $\mathbf{AX} = \mathbf{b}$ se poate scrie astfel :

$$\mathbf{X}^{\mathbf{B}} = \mathbf{B}^{-1} \cdot \mathbf{b} - \mathbf{B}^{-1} \cdot \mathbf{S} \cdot \mathbf{X}^{\mathbf{S}}.$$

Introducînd notaţiile $\bar{\mathbf{X}}^{\mathbf{B}} = \mathbf{B}^{-1} \cdot \mathbf{b}$ şi $y_j^{\mathbf{B}} = \mathbf{B}^{-1} \cdot a_j$, unde a_j reprezintă coloana j din matricea $\mathbf{A}$ corespunzătoare variabilelor secundare ($j \in \mathbf{J_S}$), unde $\mathbf{J_S}$ este mulţimea $\{j : a_j \in \mathbf{S}\}$, atunci egalitatea de mai sus devine :

$$\mathbf{X}^{\mathbf{B}} = \bar{\mathbf{X}}^{\mathbf{B}} - \sum_{j \in \mathbf{J_S}} y_j^{\mathbf{B}} x_j$$

sau

$$x_i^{\mathbf{B}} = \bar{x}_i^{\mathbf{B}} - \sum_{j \in \mathbf{J_s}} y_{ij}^{\mathbf{B}} x_j \quad (i \in \mathbf{J_B}) \tag{2.1}$$

unde $\mathbf{J_B} = \{i : a_i \in \mathbf{B}\}$.

Să vedem acum cum putem exprima funcţia obiectiv cu ajutorul variabilelor secundare $\mathbf{X}^{\mathbf{S}}$. Pentru aceasta pornim de la expresia funcţiei obiectiv şi facem înlocuirile succesive pe baza notaţiilor de mai sus :

$$Z = \mathbf{C}'\mathbf{X} = \sum_{j=1}^{m} c_j x_j = \sum_{i \in \mathbf{J_B}} c_i x_i + \sum_{j \in \mathbf{J_S}} c_j x_j =$$

$$= \sum_{i \in \mathbf{J_B}} c_i (\bar{x}_i^{\mathbf{B}} - \sum_{j \in \mathbf{J_S}} y_{ij}^{\mathbf{B}} x_j) + \sum_{j \in \mathbf{J_S}} c_j x_j = \sum_{i \in \mathbf{J_B}} c_i \bar{x}_i^{\mathbf{B}} - \sum_{j \in \mathbf{J_S}} (\sum_{i \in \mathbf{J_B}} c_i y_{ij}^{\mathbf{B}} - c_j) x_j.$$

Dacă folosim notaţiile :

$$\mathbf{Z}^{\mathbf{B}} = \sum_{i \in \mathbf{J_B}} c_i x_i^{\mathbf{B}} = \mathbf{C}_{\mathbf{B}}' \bar{\mathbf{X}}^{\mathbf{B}} \tag{2.2}$$

şi

$$\mathbf{Z}_j^{\mathbf{B}} = \sum_{i \in \mathbf{J_B}} c_i y_{ij}^{\mathbf{B}} = \mathbf{C}_{\mathbf{B}}' y_j^{\mathbf{B}}$$

se obţine relaţia :

$$\mathbf{Z} = \mathbf{Z}^{\mathbf{B}} - \sum_{j \in \mathbf{J_S}} (\mathbf{z}_j^{\mathbf{B}} - c_j) x_j \tag{2.3}$$

Scanned with OKEN Scanner

În cele ce urmează vom folosi pentru funcţia obiectiv operatorul „minim" urmînd să precizăm ce se întîmplă cînd trecem la maxim.

Pentru a pune în evidenţă algoritmul simplex vom prezenta unele propoziţii, care vor constitui fundamentele algoritmului de calcul a soluţiei optime.

Propoziţia 1. Dacă $z_j - c_j \leqslant 0$ pentru orice $j \in \mathbf{J_S}$, atunci soluţia admisibilă corespunzătoare bazei $\mathbf{B}(\mathbf{X^B} = \mathbf{B^{-1}b},\ \mathbf{X^S} = 0)$ este optimă.

Demonstraţie. Să notăm cu $\mathbf{Z}$ valoarea funcţiei obiectiv corespunzătoare acestei soluţii admisibile de bază. Pentru orice soluţie admisibilă, din faptul că $x_j \geqslant 0$ rezultă $\sum\limits_{j \in \mathbf{J_S}} (z_j - c_j)\, x_j \leqslant 0$ şi deci dacă notăm cu $\mathbf{Z}$ valoarea corespunzătoare a funcţiei obiectiv, obţinem $\mathbf{Z} = \mathbf{Z} - \sum\limits_{j \in \mathbf{J_S}} (z_j - c_j)\, x_j \geqslant \mathbf{Z}$ ceea ce justifică afirmaţia.

Propoziţia 2. Dacă există $k \in \mathbf{J_S}$ astfel încît $z_k - c_k > 0$ atunci soluţia admisibilă asociată bazei $\mathbf{B}$ nu este optimă (cu excepţia soluţiei degenerate, caz pe care îl vom analiza mai tîrziu) şi poate fi îmbunătăţită dacă a_k ia valori pozitive.

Demonstraţie. Într-adevăr dacă valoarea variabilei secundare creşte pînă la valoarea $x'_k > 0$, obţinem un nou program cu valoarea funcţiei obiectiv :

$$\mathbf{Z}^0 = \mathbf{Z} - (z_k - c_k)\, x_k^0 < \mathbf{Z}$$

adică un program mai bun (în sensul că dă funcţiei obiectiv o valoare mai mică).

Noul program obţinut nu este un program de bază deoarece are $m + 1$ componente strict pozitive x_i, $(i \in \mathbf{J_B})$ şi x_k.

Desigur atunci cînd x_k creşte se poate ca să nu mai rămînem în tronsonul programelor. Atunci se pune întrebarea cît de mult poate să crească x_k pentru a rămîne totuşi în tronsonul programelor. Ceea ce se poate constata imediat este faptul că odată cu creşterea lui x_k se modifică şi valorile variabilelor de bază în conformitate cu relaţia [vezi 2.1] :

$$x_i^{\mathbf{B}} = \bar{x}_i^{\mathbf{B}} - y_{ik}^{\mathbf{B}}\, x_k \qquad i \in \mathbf{J_B} \qquad (2.4)$$

Propoziţia 3. Dacă există $k \in \mathbf{J_S}$ astfel încît $z_k - c_k > 0$ şi toţi $y_{ik} \leqslant 0$ pentru toţi $i \in \mathbf{J_B}$, atunci problema are un optim infinit*.

Demonstraţie. Dacă toţi $y_{ik} \leqslant 0$ atunci din relaţia (2.4) se vede că variabila x_k poate să crească oricît de mult fără ca vreuna din variabilele de bază să devină negativă. Or, în acest caz funcţia obiectiv

$$\mathbf{Z} = \mathbf{Z} - (z_k - c_k)x_k \underset{x_k \to +\infty}{\longrightarrow} -\infty$$

* Pentru simplificarea scrierii se renunţă la indicele B care pune în evidenţă faptul că elementele $\mathbf{Z^B}$, $z_j^{\mathbf{B}}$, $y_{ij}^{\mathbf{B}}$, $\bar{\mathbf{X}}^{\mathbf{B}}$ şi $\mathbf{C_B}$ sînt exprimate în baza $\mathbf{B}$.

Scanned with OKEN Scanner

aşadar funcţia obiectiv tinde către $-\infty$ de-a lungul razei.

$$x_i = \bar{x}_i - y_{ik}\, x_k \;, \quad i \in \mathbf{J_B}$$
$$x_j = 0 \qquad\qquad , \; j \in \mathbf{J_B} - \{k\}\,; \; x_k \geqslant 0$$

Propoziţia 4. Dacă $z_k - c_k > 0$ şi există $y_{ik} > 0$ atunci valoarea variabilei x_k poate să crească pînă atinge valoarea θ_l :

$$\theta_l = \frac{\bar{x}_l}{y_{lk}} = \min_{\substack{i \in \mathbf{J_B} \\ y_{ik} > 0}} \left(\frac{\bar{x}_i}{y_{ik}} \right) \tag{2.5}$$

în care caz se obţine un nou program de bază, asociat bazei $\tilde{\mathbf{B}}$ dedusă din $\mathbf{B}$ prin înlocuirea vectorului a_k cu vectorul a_l.

Demonstraţie. Dacă, $y_{ik} > 0$, atunci pentru a avea $x_i = \bar{x}_i - y_{ik}\, x_k \geqslant 0$ pentru toţi $i \in \mathbf{J_B}$ trebuie ca :

$$x_k \leqslant \frac{x_i}{y_{ik}}$$

pentru toţi $i \in \mathbf{J_B}$, $y_{ik} > 0$.

Să presupunem că valoarea variabilei x_k poate creşte numai pînă la valoarea rezultată din (2.5) şi că acest minim este atins pentru indicele $l \in \mathbf{J_B}$. Pentru această valoare, x_l se anulează şi deci obţinem un nou program de bază în care variabilele de bază sînt x_k şi x_i ($i \in \mathbf{J_B} - \{k\}$) adică un program corespunzător bazei $\tilde{\mathbf{B}}$ obţinute din $\mathbf{B}$ prin înlocuirea vectorului a_l cu vectorul a_k. Din faptul că $y_{ik} \neq 0$ rezultă că vectorii coloană din matricea $\tilde{\mathbf{B}}$ sînt liniar independenţi.

Aşadar, x_k nu poate creşte mai mult decît valoarea rezultată din (2.5) căci atunci ar deveni negativă valoarea variabilei x_l. Corespunzător acestei creşteri, valoarea funcţiei obiectiv scade cu cantitatea

$$\frac{x_l}{y_{lk}} (z_k - c_k)$$

Să remarcăm faptul că este posibil să existe mai mulţi indici pentru care $z_j - c_j > 0$. Dintre aceştia este preferabil să se aleagă acela care maximizează expresia $\frac{x_l}{y_{lj}} (z_j - c_j)$ deoarece aceasta asigură cea mai rapidă scădere a funcţiei obiectiv şi va conduce după cum se va constata, la un număr mai mic de iteraţii în algoritmul simplex.

În practică se foloseşte totuşi un criteriu mai simplu care se manevrează mai uşor şi anume : se alege acel indice k pentru care $z_j - c_j$ este maximă, adică $z_k - c_k = \max_j (z_j - c_j)$.

Prin urmare aplicarea algoritmului simplex la rezolvarea unei probleme de programare liniară constă în următoarele etape :

Scanned with OKEN Scanner

a) Se determină o bază inițială B și pentru această bază se calculează mărimile :

$$\bar{x}_i^{\mathbf{B}},\ \mathbf{Z}^{\mathbf{B}},\ z_j^{\mathbf{B}} - c_j,\ y_{ij}^{\mathbf{B}}$$

b) Se analizează diferențele $z_j - c_j$ (criteriul de intrare în bază).

b_1) dacă toate diferențele $z_j - c_j \leqslant 0$, programul este optim ;

b_2) dacă există cel puțin un indice $j \in \mathbf{J_S}$ astfel încît $z_j - c_j > 0$ se determină $k \in \mathbf{J_S}$ pentru care $z_k - c_k = \max\limits_j (z_j - c_j)$.

c) Se stabilește vectorul ce urmează a ieși din bază (criteriul de ieșire din bază).

c_1) dacă toți $y_{ik} \leqslant 0$ problema are optim infinit ;

c_2)dacă există $y_{ik} > 0$ se determină indicele l astfel încît să fie satisfăcută relația (2.5).

d) Se înlocuiește în baza inițială $\mathbf{B}$ vectorul a_l cu vectorul a_k obținîndu-se o nouă bază $\tilde{\mathbf{B}}$, ceea ce înseamnă că se transformă mărimile $\bar{x}_i$, $\bar{\mathbf{Z}}$, $z_j - c_j$, $y_{ij}^{\mathbf{B}}$ după relațiile :

$$\bar{x}_k^{\tilde{\mathbf{B}}} = \frac{x_l}{y_{lk}};\ y_{kj}^{\tilde{\mathbf{B}}} = \frac{y_{lj}}{y_{lk}} \tag{2.6}$$

$$\bar{x}_i^{\tilde{\mathbf{B}}} = x_i - y_{ik}\frac{\bar{x}_l}{y_{lk}};\ y_{ij}^{\tilde{\mathbf{B}}} = y_{ij} - y_{ik}\frac{y_{lj}}{y_{lk}},\ i \neq l \tag{2.7}$$

$$\mathbf{Z}^{\tilde{\mathbf{B}}} = \mathbf{Z}^{\mathbf{B}} - (z_k - c_k)\frac{\bar{x}_l}{y_{lk}};\ z_j^{\tilde{\mathbf{B}}} - c_j = (z_j - c_j) - (z_k - c_k)\frac{y_{lj}}{y_{lk}} \tag{2.8}$$

și se reia aplicarea algoritmului de la pasul (b).

Înainte de a trece mai departe să justificăm relațiile (2.6), (2.7) și (2.8) de mai sus.

Din (2.1) obținem pentru $i = l$

$$x_l = \bar{x}_l - \sum_{j \in \mathbf{J_S} - \{k\}} y_{lj}x_j - y_{lk}x_k$$

Din această relație se determină :

$$x_k = \frac{1}{y_{lk}}\left[\bar{x}_l - \sum_{j \in \mathbf{J_S} - \{k\}} y_{lj}x_j - x_l\right] = \bar{x}_k^{\tilde{\mathbf{B}}} - \sum_{j \in \mathbf{J_S}} y_{kj}^{\tilde{\mathbf{B}}}x_j \tag{2.9}$$

unde am notat :

$$\mathbf{J_S} = \mathbf{J_S} - \{k\} + \{l\},\ \bar{x}_k^{\tilde{\mathbf{B}}} = \frac{\bar{x}_l}{y_{lk}},\ y_{kj}^{\tilde{\mathbf{B}}} = \frac{y_{lj}}{y_{lk}} \text{ (cu } y_{ll} = 1)$$

Componentele relației (2.9) sînt identice cu termenii definiți prin (2.6) și reprezintă noile elemente ale rîndului $i = l$ exprimate în baza $\mathbf{B}$.

Scanned with OKEN Scanner

Pentru a exprima acum celelalte variabile de bază x_i, $i \in J_{\tilde{B}} - \{k\}$ în funcţie de variabilele secundare x_j, $j \in J_{\tilde{S}}$ înlocuim x_k obţinut mai sus în celelalte relaţii ce ne dau pe x_i, $i \in J_{\tilde{B}} - \{l\}$.

Avem aşadar

$$x_i = \bar{x}_i - \sum_{j \in J_S - \{k\}} y_{ij} x_j - y_{ik} x_k =$$

$$= \bar{x}_i - \sum_{j \in J_S - \{k\}} y_{ij} x_j - \frac{y_{ik}}{y_{lk}} \left(\bar{x}_l - \sum_{j \in J_S - \{k\}} y_{lj} x_j - x_l \right) =$$

$$= \left(\bar{x}_i - y_{ik} \frac{\bar{x}_l}{y_{lk}} \right) - \sum_{j \in J_S - \{k\}} \left(y_{ij} - y_{ik} \frac{y_{lj}}{y_{lk}} \right) x_j + \frac{y_{ik}}{y_{lk}} x_l =$$

$$= \bar{x}_i^{\tilde{B}} - \sum_{j \in J_{\tilde{S}}} y_{ij}^{\tilde{B}} \cdot x_j$$

unde

$$\bar{x}_i^{\tilde{B}} = \bar{x}_i - y_{ik} \frac{\bar{x}_l}{y_{lk}} \quad \text{şi} \quad y_{ij}^{\tilde{B}} = y_{ij} - y_{ik} \frac{y_{lj}}{y_{lk}} \quad (i \neq l)$$

Termenii acestei relaţii sînt tocmai cei definiţi prin (2.7).

În fine dacă pornim de la relaţia (2.3) şi înlocuim cu noile variabile secundare se obţine:

$$Z = \bar{Z} - \sum_{j \in J_S - \{k\}} (z_j - c_j) x_j - (z_k - c_k) x_k =$$

$$= \bar{Z} - \sum_{j \in J_S - \{k\}} (z_j - c_j) x_j - (z_k - c_k) \frac{1}{y_{lk}} \left[\bar{x}_l - \sum_{j \in J_S - \{k\}} y_{lj} x_j - x_l \right] =$$

$$= \bar{Z}^{\tilde{B}} - \sum_{j \in J_{\tilde{S}}} (z_j^{\tilde{B}} - c_j) x_j$$

unde am notat cu: $J_{\tilde{S}} = J_S - \{k\} + \{l\}$, $\bar{z}^{\tilde{B}} = \bar{z}^B - (z_k - c_k) \dfrac{x_l}{y_{lk}}$.

$$Z_j^{\tilde{B}} - c_j = z_j - c_j - (z_k - c_k) \frac{y_{lj}}{y_{lk}}.$$

Relaţia obţinută justifică formulele (2.8). Elementul situat la intersecţia rîndului l cu coloana k se numeşte pivot.

Scanned with OKEN Scanner

Pentru o problemă în care se cere maximizarea funcției $\mathbf{Z}(x)$ în enunțul algoritmului simplex se modifică numai criteriul de intrare în bază și anume :

(b') $\begin{cases} b_1') \text{ dacă toate diferențele } z_j - c_j \geqslant 0 \text{ atunci programul este optim ;} \\ b_2') \text{ dacă există cel puțin un indice } j \in \mathbf{J_S} \text{ astfel încît } z_j - c_j < 0 \\ \quad \text{se determină } k \text{ astfel încît } z_k - c_k = \min\limits_j (z_j - c_j). \end{cases}$

Elementele necesare aplicării algoritmului simplex se trec într-un tabel numit tabel simplex. Pentru simplificarea scrierii am presupus că baza $\mathbf{B}$ este formată din primele m coloane ale matricii $\mathbf{A}$ și de aceea tabelul simplex inițial are forma prezentată în tabelul 2.6.

Dacă pentru o problemă de minim există diferențe $\Delta z_j = z_j - c_j > 0$ (iar pentru cele de maxim $\Delta z_j = z_j - c_j < 0$) soluția $\mathbf{X^B} = (\bar{x}_1, \bar{x}_2, \ldots \bar{x}_m, 0, \ldots, 0)$ nu este optimă și va intra în bază vectorul a_k determinat cu ajutorul criteriului de intrare $\Delta z_k = \max |z_j - c_j|$. Vectorul a_k va intra în bază în locul vectorului a_l stabilit prin aplicarea criteriului de ieșire din bază.

$$\theta_l = \min_{i \in \mathbf{J_B}} \left(\frac{\bar{x}_i}{y_{ik}} \right) = \frac{\bar{x}_l}{y_{lk}}$$

$$y_{ik} > 0$$

Precizăm că în tabelul simplex inițial $y_{ij} = a_{ij}$ și ca urmare pivotul este elementul a_{lk}.

Pasul următor al algoritmului simplex constă în aplicarea relațiilor (2.6), (2.7) și (2.8) cu ajutorul cărora calculăm elementele tabelului simplex corespunzător bazei $\tilde{\mathbf{B}}$. Toate calculele prezentate mai sus, necesare obținerii unei noi baze, constituie o iterație a algoritmului simplex.

Exemplul 2.5. În continuare vom aplica algoritmul simplex la rezolvarea primei probleme prezentate în paragraful 2.1.

Prin introducerea variabilelor de ecart x_3 și x_4 se obține forma standard :

$$\max [\mathbf{Z}(x)] = 10x_1 + 15x_2$$
$$2x_1 + 4x_2 + x_3 = 40$$
$$6x_1 + 2x_2 + x_4 = 60$$
$$x_j \geqslant 0 \ (j = 1, 2, 3, 4)$$

Problema analizată are un program de bază care se determină imediat, deoarece matricea

$$\mathbf{A} = \begin{pmatrix} 2 & 4 & 1 & 0 \\ 6 & 2 & 0 & 1 \end{pmatrix}$$

conține baza $\mathbf{B}$:

$$\mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Tabelul 2.6

B	C_B	$\overline{X}_B$	c_1	$c_2 \dots c_l$	$\dots c_m$	$c_{m+1} \dots$	$c_k \dots c_n$	θ_i
			a_1	$a_2 \dots a_l$	$\dots a_m$	$a_{m+1} \dots$	$a_k \dots a_n$	
a_1	c_1	$\bar{x}_1$	1	$0 \dots 0$	$\dots 0$	$a_{1,m+1} \dots$	$a_{1k} \dots a_{1n}$	$\frac{\bar{x}_1}{a_{1k}}$
a_2	c_2	$\bar{x}_2$	0	$1 \dots 0$	$\dots 0$	$a_{2,m+1} \dots$	$a_{2k} \dots a_{2n}$	$\frac{\bar{x}_2}{a_{2k}}$
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
a_l	c_l	$\bar{x}_l$	0	$0 \dots 1$	$\dots 0$	$a_{l,m+1} \dots$	$a_{lk} \dots a_{ln}$	$\frac{\bar{x}_l}{a_{lk}}$
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
a_m	c_m	$\bar{x}_m$	0	$0 \dots 0$	$\dots 1$	$a_{m,m+1} \dots$	$a_{mk} \dots a_{mn}$	$\frac{\bar{x}_m}{a_{mk}}$
z_j		$\overline{Z}$	c_1	$c_2 \dots c_1$	$\dots c_n$	$z_{m+1} \dots$	$z_k \dots z_n$	—
Δz_j		—	0	$0 \dots 0$	$\dots 0$	$\Delta z_{m+1} \dots$	$\Delta z_k \dots \Delta z_n$	—

Observație: Precizăm că în locul vectorilor a_j se trec adesea variabilele x_j. Pe ultimul rînd sînt trecute diferențele $\Delta z_j = z_j - c_j$ corespunzătoare vectorilor a_j.

Scanned with OKEN Scanner

Prin urmare soluţia admisibilă* de start are structura $\bar{X}^B = (0,0, 40, 60)$ iar valoarea funcţiei obiectiv corespunzătoare este $\bar{Z}^B(x)=0$. În tabelul 2.7 se prezintă tabelul simplex iniţial T_0 precum şi rezultatele celor două iteraţii necesare pentru rezolvarea problemei date (tabelul T_1 şi T_2).

Elementele $\bar{Z}^B$, z_j^B şi ΔZ_j ale tabelului T_0 se calculează pe baza relaţiilor (2.2):

$$\bar{Z}^B = \sum_{i \in J_B} c_i \bar{x}_i = 0 \times 40 + 0 \times 60 = 0;$$

$$z_1^B = \sum_{i \in J_B} c_i a_{i1} = 0 \times 2 + 0 \times 6 = 0; \quad z_2^B = \sum_{i \in J_B} c_i a_{i2} = 0$$

$$\Delta z_1 = z_1 - c_1 = 0 - 10 = -10; \quad \Delta z_2 = z_2 - c_2 = -15.$$

Tabelele 2.7

	B	C_B	$\bar{X}$	10	15	0	0	θ_i
				a_1	a_2	a_3	a_4	
T_0	a_3	0	40	2	④	1	0	40/4 = 10
	a_4	0	60	6	2	0	1	60/2 = 30
	z_j		0	0	0	0	0	—
	Δz_j		—	−10	−15	0	0	—
T_1	a_2	15	10	1/2	1	1/4	0	20
	a_4	0	40	⑤	0	−1/2	1	8
	z_j		150	15/2	15	15/4	0	—
	Δz_j		—	−5/2	0	15/4	0	—
T_2	a_2	15	6	0	1	3/10	−1/10	
	a_1	10	8	1	0	−1/10	1/5	
	z_j		170	10	15	7/2	1/2	
	Δz_j		—	0	0	7/2	1/2	

* Întrucît vectorii coloană a_1 şi a_2 sînt liniar independenţi [det $(a_1, a_2) = -20$] aceştia formează o bază. Avînd în vedere că:

$$B^{-1} = \begin{pmatrix} -1/10 & 1/15 \\ 3/10 & -1/10 \end{pmatrix}$$

rezultă că soluţia corespunzătoare bazei (a_1, a_2) este

$$\bar{X}^B = \begin{pmatrix} -1/10 & 1/5 \\ 3/10 & -1/10 \end{pmatrix} \cdot \begin{pmatrix} 40 \\ 60 \end{pmatrix} = \begin{pmatrix} 8 \\ 6 \end{pmatrix}$$

Componentele vectorilor a_3 şi a_4 în această bază se calculează din relaţia $y_j = B^{-1} a_j$ $(j = 3,4)$ obţinîndu-se datele din T_2.

Însă pentru problemele mari este mai comod să se înceapă cu o bază formată dintr-un sistem de m vectori unitari.

Scanned with OKEN Scanner

Rezultă că soluția din tabelul T_0 nu este optimă deoarece există diferențe $\Delta z_j < 0$ și de aceea se introduce în bază vectorul a_2 întrucît :

$$\min (\Delta z_1, \Delta z_2) = \Delta z_2$$

Pentru a stabili ce vector urmează a ieși din bază $\mathbf{B}$ se calculează raporturile $\theta_i = x_i/a_{i2}$ (pentru $a_{i2} > 0$) și se aplică criteriul de ieșire (2.5) :

$$\min (40/4,\ 60/2) = 10.$$

Raportul minim, $\theta_1 = 10$, corespunde vectorului a_3 care iese din bază și în locul lui va intra vectorul a_2. Dispunînd de o nouă bază $\tilde{\mathbf{B}} = (a_2, a_4)$ se elaborează tabelul T_2 utilizînd relațiile (2.6) și (2.7) pentru a exprima componentele vectorilor în această bază. Elementele tabelului T_2 s-au calculat astfel :

a) Toate elementele primului rînd au rezultat prin împărțirea la pivot ($a_{12} = 4$) a elementelor corespunzătoare din tabelul T_0 [vezi relația (2.6)]. Spre exemplu :

$$\bar{x}_2^{\tilde{\mathbf{B}}} = \frac{\bar{x}_2}{a_{12}} = \frac{40}{4} = 10\ ;\ y_{11}^{\tilde{\mathbf{B}}} = \frac{a_{11}}{a_{12}} = \frac{2}{4} = \frac{1}{2}$$

b) Elementele celui de-al doilea rînd ($i = 2$) se calculează pe baza relației (2.7) astfel :

$$\bar{x}_4^{\tilde{\mathbf{B}}} = x_4 - \frac{\bar{x}_1}{a_{12}} a_{22} = 60 - \frac{40}{4} \cdot 2 = 40$$

$$y_{21}^{\tilde{\mathbf{B}}} = a_{21} - \frac{a_{11}}{a_{12}} a_{22} = 6 - \frac{1}{2} \cdot 2 = 5$$

$$y_{23}^{\tilde{\mathbf{B}}} = a_{23} - \frac{a_{13}}{a_{12}} a_{22} = 0 - \frac{1}{4} \cdot 2 = -\frac{1}{2}$$

c) Indicatorii $\bar{\mathbf{Z}}^{\tilde{\mathbf{B}}}$ și $z_j^{\tilde{\mathbf{B}}} - c_j$ se calculează cu ajutorul relațiilor :

$$\bar{\mathbf{Z}}^{\mathbf{B}} = \sum_{i \in J_{\tilde{\mathbf{B}}}} c_i \bar{x}_i^{\tilde{\mathbf{B}}} = 15 \times 10 + 0 \times 40 = 150$$

$$\text{sau } \bar{\mathbf{Z}}^{\tilde{\mathbf{B}}} = \bar{\mathbf{Z}}^{\mathbf{B}} - (z_2 - c_2)\frac{\bar{x}_3}{a_{12}} = 0 + 15\,\frac{40}{4} = 150$$

$$z_1^{\tilde{\mathbf{B}}} = \sum_{i \in J_{\tilde{\mathbf{B}}}} c_i y_{i1} = 15 \cdot \frac{1}{2} + 0 \cdot 5 = \frac{15}{2};$$

$$z_3^{\tilde{\mathbf{B}}} = \sum_{i \in J_{\tilde{\mathbf{B}}}} c_i y_{i3} = 15 \cdot \frac{1}{4} - 0 \cdot \frac{1}{2} = \frac{15}{4}$$

$$\Delta z_1^{\tilde{\mathbf{B}}} = z_1^{\tilde{\mathbf{B}}} - c_1 = \frac{15}{2} - 10 = -\frac{5}{2};\ \Delta z_3^{\tilde{\mathbf{B}}} = z_3^{\tilde{\mathbf{B}}} - c_3 = \frac{15}{4} - 0 = \frac{15}{4}$$

Avînd în vedere că singura diferenţă negativă este $\Delta z_1^{\tilde{B}} = -\frac{5}{2}$, va intra în bază vectorul a_1 în locul vectorului a_4, deoarece $\min(20, 8) = 8$. Calculele necesare obţinerii tabelului T_2 sînt identice cu cele descrise mai înainte. Întrucît în acest tabel toate diferenţele sînt negative, soluţia $\bar{x} = (8,\ 6,\ 0,\ 0)$ este optimă, realizîndu-se o valoare a funcţiei obiectiv egală cu *170* u.m.

2.6. Convergenţa algoritmului simplex. Degenerarea în programarea liniară

Prin aplicarea algoritmului simplex se obţine, într-un număr finit de cel mult C_n^m iteraţii, soluţia problemei de programare liniară care poate fi un optim finit sau infinit. Dar în prezentarea făcută în capitolele precedente s-a considerat ipoteza că soluţiile corespunzătoare diferitelor baze sînt nedegenerate. După cum se va vedea în continuare, soluţiile degenerate implică adaptarea algoritmului simplex în vederea eliminării degenerării.

Înainte de a discuta aspectele legate de utilizarea algoritmului simplex pentru problemele degenerate, vom analiza cazurile de degenerare şi consecinţele acestora asupra modelului de programare liniară. Pot apărea programe degenerate în următoarele situaţii:

a) Dacă vectorul termenilor liberi are unele componente nule, ceea ce înseamnă că acesta este situat în planul determinat de unii dintre vectorii a_j, atunci la aplicarea criteriului de ieşire din bază raportul $\theta_l = \min(\theta_1, \theta_2, \ldots, \theta_m)$ nu este unic determinat.

b) În cazul cînd vectorul termenilor liberi nu este liniar independent faţă de orice sistem format din $(m-1)$ vectorii coloană din matricea $\mathbf{A}$ (aceasta înseamnă că vectorul a_0 trece printr-un vîrf al poliedrului convex definit de restricţiile problemei), la aplicarea criteriului de ieşire din bază raportul θ_l nu este unic determinat.

Pentru o problemă care are cel puţin un program degenerat există posibilitatea să apară, în afară de consecinţele enunţate, mai sus, fenomenul de ciclare care constă în revenirea la una dintre bazele corespunzătoare unei iteraţii anterioare. În această situaţie valoarea funcţiei obiectiv rămîne nemodificată pe parcursul unor iteraţii şi ca urmare procesul iterativ continuă la infinit.

Trebuie precizat însă că degenerarea nu conduce neapărat la fenomenul de ciclare şi că în problemele practice rezolvate pînă în prezent nu s-a constatat apariţia ciclării.

În vederea eliminării degenerării se foloseşte tehnica perturbării* care constă în înlocuirea sistemului de restricţii a problemei de programare liniară dată sub forma standard, cu sistemul :

$$\mathbf{Ax} = \mathbf{b} + \sum_{j=1}^{n} a_j \varepsilon^j \tag{2.10}$$

unde a_j reprezintă vectorii coloană a matricei **A**, iar $\varepsilon > 0$ este un număr arbitrar suficient de mic**.

Fără a restrînge generalitatea, putem presupune că dispunem de baza $\mathbf{B} = (a_1, a_2, \ldots, a_m)$. Prin înmulţirea la stînga a relaţiei (2.10) cu $\mathbf{B}^{-1}$ se obţine soluţia corespunzătoare acestei baze :

$$\bar{x}_i^{\mathbf{B}}(\varepsilon) = \bar{x}_i^{\mathbf{B}} + \varepsilon^i + \sum_{j=m+1}^{n} y_{ij}^{\mathbf{B}} \varepsilon^j \quad (i = 1, 2, \ldots, m)$$

sau

$$\bar{x}_i^{\mathbf{B}}(\varepsilon) = \bar{x}_i^{\mathbf{B}} + \sum_{j=1}^{n} y_{ij}^{\mathbf{B}} \varepsilon^j \quad (i = 1, 2, \ldots, m)$$

Cele două relaţii sînt echivalente deoarece $y_{ii} = 1$, iar celelalte elemente ale rîndului i, $y_{ij} = 0$ pentru toţi $j \in \mathbf{B}$.

Datorită perturbării vectorului termenilor liberi se modifică criteriul de ieşire din bază, vectorul ce urmează a fi eliminat determinîndu-se din relaţia :

$$\min_{i(y_{ik} > 0)} \left(\frac{x_i^{\mathbf{B}}(\varepsilon)}{y_{ik}^{\mathbf{B}}} \right) = \min_{i, y_{ik} > 0} \left(\frac{\bar{x}_i}{y_{ik}} + \frac{y_{i1}}{y_{ik}} \varepsilon + \frac{y_{i2}}{y_{ik}} \varepsilon^2 + \ldots + \frac{y_{in}}{y_{ik}} \varepsilon^n \right)$$

în acest fel problema perturbată nu mai este degenerată.

Pe baza relaţiei date mai sus se poate formula următoarea regulă practică de eliminare a degenerării :

a) se compară mai întîi rapoartele $\frac{\bar{x}_i}{y_{ik}}$ care dau dealtfel şi ordinul de mărime al rapoartelor

$$\frac{\bar{x}_i^{\mathbf{B}}(\varepsilon)}{y_{ik}} \quad i = 1, 2, \ldots$$

şi se determină minimul lor.

---

* Acest procedeu poate fi utilizat şi pentru rezolvarea problemei ciclării, dacă aceasta apare.

** Numărul ε se alege suficient de mic în aşa fel ca pentru orice $p < q$ să avem $\varepsilon^p > P_q(\varepsilon)$ unde $P_q(\varepsilon)$ este un polinom ce conţine pe ε la puteri mai mari sau egale cu q.

Scanned with OKEN Scanner

Dacă minimul nu este unic, ci se realizează pe o mulţime de indici I_0, comparăm rapoartele

$$\frac{y_{i1}}{y_{ik}} \quad i \in I_0$$

corespunzătoare puterii întîi a lui ε.

b) Dacă minimul acesta nu este unic, ci se realizează pentru o mulţime de indici I_1, se compară rapoartele :

$$\frac{y_{i2}}{y_{ik}}, \quad i \in I_1 \subset I_0$$

care sînt coeficienţii lui ε^2.

Se procedează tot aşa mai departe pînă ce obţinem un minim unic. În cel mai rău caz minimul rapoartelor :

$$\frac{y_{im}}{y_{ik}} \quad i \in I_{m-1} \subset I_{m-2}$$

trebuie să fie unic şi deci se defineşte în mod univoc variabila care iese din bază.

Aşadar dacă se modifică algoritmul simplex dat iniţial cu rezultatele obţinute aici va fi asigurată şi convergenţa algoritmului simplex (altfel spus se obţine soluţia optimă după un număr finit de iteraţii).

Exemplul 2.6. Am luat acest exemplu deoarece apar aici mulţimile de indici I_0 şi I_1 formate din mai mulţi indici, ceea ce ne dă posibilitatea să urmărim mai bine tehnica de perturbare menţionată.

Să se determine programul optim pentru următoarea problemă de programare liniară :

$$\min [Z_{(X)}] = -\frac{1}{2}x_5 + x_6 - \frac{3}{2}x_7 + 2x_8$$

În condiţiile :

$$x_1 - 2x_5 + 3x_6 + 4x_7 + x_8 = 0$$

$$x_2 - x_5 - 2x_6 + 3x_7 - x_8 = 0$$

$$x_3 + x_5 - x_6 + 2x_7 - 3x_8 = 0$$

$$x_4 + x_7 - x_8 = 2.$$

$$x_j \geqslant 0 \qquad (j = 1, 2, \ldots, 8)$$

Tabelul simplex iniţial are structura :

Baza	c_i	$\bar{x}$	0	0	0	0	−1/2	1	−3/2	2
			a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
a_1	0	0	1	0	0	0	−2	3	4	1
a_2	0	0	0	1	0	0	−1	−2	3	−1
a_3	0	0	0	0	1	0	1	−1	②	−3
a_4	0	2	0	0	0	1	0	0	1	−1
z_j	—	$\bar{z}=0$	0	0	0	0	0	0	0	0
$z_j - c_j$	—	—	0	0	0	0	1/2	−1	3/2	−2

Întrucît există diferenţe $z_j - c_j > 0$ înseamnă că soluţia nu este optimă. Constatăm că $\max\limits_{j=5,7}(z_j - c_j) = \max(1/2,\ 3/2) = 3/2$ şi deci va intra în noua bază vectorul a_7. Pentru a vedea ce vector iese din bază aflăm :

$$\min\left(\frac{\bar{x}_i}{y_{i7}}\right) = \min\left(\frac{\bar{x}_1}{y_{17}}, \frac{\bar{x}_2}{y_{27}}, \frac{\bar{x}_3}{y_{37}}, \frac{\bar{x}_4}{y_{47}}\right) = \min\left(\frac{0}{4}, \frac{0}{3}, \frac{0}{2}, \frac{2}{1}\right) = 0$$

Minimul acesta nu este unic determinat, ci se realizează după mulţimea de indici $I_0 = \{1, 2, 3\}$. Aplicăm tehnica de perturbare descrisă mai înainte şi obţinem :

$$\min_{i \in I_0}\left(\frac{y_{i1}}{y_{i7}}\right) = \min\left(\frac{y_{11}}{y_{17}}, \frac{y_{21}}{y_{27}}, \frac{y_{31}}{y_{37}}\right) = \min\left(\frac{1}{4}, \frac{0}{3}, \frac{0}{2}\right) = 0$$

Nici acest minim nu este unic determinat ci se realizează după mulţimea de indici $I_1 = \{2, 3\}$

Aflăm

$$\min_{i \in I_1}\left(\frac{y_{i2}}{y_{i7}}\right)$$

care în cazul nostru este

$$\min_{i \in I_1}\left(\frac{y_{i2}}{y_{i7}}\right) = \min\left(\frac{y_{22}}{y_{27}}, \frac{y_{32}}{y_{37}}\right) = \min\left(\frac{1}{3}, \frac{0}{2}\right) = 0$$

şi deci $l = 3$, adică iese din bază vectorul a_3 locul lui fiind luat de a_7.

2.7. Determinarea unui program de bază iniţial Metoda celor două faze Metoda bazei artificiale

Aşa după cum s-a văzut, pentru a se putea aplica algoritmul simplex este nevoie să se cunoască un program de bază.

Scanned with OKEN Scanner

Dacă modelul matematic al problemei de programare liniară este de forma

$$\max\ [\mathbf{C'X}]$$
$$\mathbf{AX} \leqslant \mathbf{b}$$
$$\mathbf{X} \geqslant 0$$

aducîndu-l la forma standard apare totdeauna un program de bază şi anume programul corespunzător bazei formată din vectorii auxiliari. Cum nu toate problemele ce intervin în practică sînt de această formă rezultă că este necesar să cunoaştem metode generale pentru determinarea unui program de bază iniţial.

Dat fiind faptul că algoritmul simplex se aplică pe forma standard vom considera problema de programare liniară dată sub forma :

$$\min\,(\max)\ [\mathbf{Z}(x)] = \mathbf{C'X}$$
$$\mathbf{AX} = \mathbf{b} \tag{2.11}$$
$$\mathbf{X} \geqslant 0$$

Pentru determinarea unui program de bază care să constituie punctul de plecare al algoritmului simplex noi vom aborda metoda celor două faze şi metoda bazei artificiale.

2.7.1. Metoda celor două faze

Vom presupune că matricea **A** a problemei de programare liniară (2.11) nu conţine nici un vector unitar şi că $\mathbf{b} \geqslant 0$ (în caz contrar n-avem decît să înmulţim cu -1 restricţiile pentru care $\mathbf{b}_i < 0$).

Dacă se adugă fiecărei ecuaţii (restricţii) „i" cîte o „variabilă artificială", „x_i^a" $\geqslant 0$ se obţine

$$\mathbf{AX} + \mathbf{I}\,\mathbf{X}^a = \mathbf{b}$$
$$\mathbf{X} \geqslant 0,\ \mathbf{X}^a \geqslant 0 \tag{2.12}$$

unde **I** este matricea unitate de tipul $m \times m$. Se constată imediat că problema (2.11) are soluţie dacă şi numai dacă sistemul (2.12) are soluţii de forma $(\mathbf{X},\ \mathbf{X}^a) \geqslant 0$ cu $\mathbf{X}^a = 0$.

Dacă obţinem o astfel de soluţie de bază cu $\mathbf{X}^a = 0$, atunci vectorul **X** constituie o soluţie de bază iniţială pentru problema (2.11).

Aşadar pentru a obţine o soluţie a sistemului (2.12) cu $\mathbf{X}^a = 0$, rezolvăm problema de programare liniară :

$$\min \mathbf{W} = \min \sum_{i=1}^{m} x_i^a$$
$$\mathbf{AX} + \mathbf{IX}^a = \mathbf{b}$$
$$\mathbf{X} \geqslant 0,\ \mathbf{X}^a \geqslant 0$$

Scanned with OKEN Scanner

folosind algoritmul simplex obişnuit. Aceasta constituie faza I a metodei şi are ca obiectiv eliminarea variabilelor artificiale.

Pentru această problemă de programare liniară dispunem, după cum se vede uşor, de soluţia de bază iniţială $\mathbf{X}=0$, $\mathbf{X}^a = b$ corespunzătoare bazei definite de matricea $\mathbf{I}$ ataşată sistemului de restricţii.

Deoarece $x_i^a \geqslant 0$, rezultă că $\sum_{i=1}^{m} x_i^a \geqslant 0$ şi deci şi $\min \sum_{i=1}^{m} x_i^a \geqslant 0$.

Sînt posibile acum două situaţii :

1) Sau $\min \mathbf{W} > 0$ şi atunci problema (2.11) nu are soluţii întrucît nu toate variabilele artificiale s-au eliminat din bază.

2) Sau toate variabilele artificiale au fost eliminate din bază, deci $\min \mathbf{W} = 0$ şi atunci am obţinut o soluţie cu $x_i^a = 0$ $(1 \leqslant i \leqslant m)$ deci un program de bază* al problemei (2.11) şi cu acest program de bază se trece la rezolvarea problemei

$$\min[\mathbf{C}'\mathbf{X}]$$

$$\mathbf{AX} = \mathbf{b}$$

$$\mathbf{X} \geqslant 0$$

care constituie faza a doua a metodei (ne-am fixat asupra minimului la funcţia obiectiv).

Dacă în problema (2.11) există o variabilă x_j care apare într-o singură ecuaţie i cu coeficientul $a_{ij} = 1$, ceea ce înseamnă că matricea $\mathbf{A}$ conţine un vector unitar, atunci în ecuaţia respectivă nu mai este necesară adăugarea unei variabile artificiale, deoarece variabila x_j poate fi luată în baza iniţială.

Se înţelege că în situaţia cînd există k astfel de variabile vom introduce doar $m-k$ variabile artificiale, iar acest lucru scurtează cu atît mai mult faza întîi, cu cît k este mai mare.

Exemplul 2.7 Să se determine soluţia optimă a următoarei probleme :

$$\max \ [\mathbf{Z}(x)] = x_1 + 2x_2 + 3x_3$$

$$3x_1 + 2x_2 + x_3 = 15$$

$$x_1 \leqslant 10$$

$$x_2 \leqslant 5$$

$$x_3 \leqslant 3$$

$$x_1, x_2, x_3 \geqslant 0$$

---

* Este posibil să apară situaţia în care $\min \mathbf{W} = 0$ dar în bază au rămas unele variabile artificiale $x_i^a = 0$. În acest caz soluţia este degenerată şi pentru a obţine o soluţie de bază a problemei (2.11) se continuă procesul de eliminare a variabilelor artificiale.

Scanned with OKEN Scanner

Vom aduce mai întîi problema la forma standard:

$$\max\ [\mathbf{Z}(x)] = x_1 + 2x_2 + 3x_3$$
$$3x_1 + 2x_2 + x_3 = 15$$
$$x_1 + x_4 = 10$$
$$x_2 + x_5 = 5$$
$$x_3 + x_6 = 3$$
$$x_j \geqslant 0 \quad (j = 1,2,\ldots,6)$$

Scriind matricea coeficienţilor tehnologici pentru problema adusă la forma standard

$$\mathbf{A} = \begin{pmatrix} 3 & 2 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix},$$

se constată că această matrice conţine trei vectori unitari şi deci trebuie să adăugăm o singură variabilă artificială (x_7). Atunci vom avea de rezolvat următoarele probleme de programare liniară:

Faza I

$$\min\ \mathbf{W} = x_7$$
$$3x_1 + 2x_2 + x_3 + x_7 = 15$$
$$x_1 + x_4 = 10$$
$$x_2 + x_5 = 5$$
$$x_3 + x_6 = 3$$
$$x_j \geqslant 0 \quad (j = 1, 2,\ldots,7)$$

Faza a II-a

$$\max\ [\mathbf{Z}(x)] = x_1 + 2x_2 + 3x_3$$
$$3x_1 + 2x_2 + x_3 = 15$$
$$x_1 + x_4 = 10$$
$$x_2 + x_5 = 5$$
$$x_3 + x_6 = 3$$
$$x_j \geqslant 0 \quad (j = 1, 2,\ldots,6)$$

Să rezolvăm problema corespunzătoare primei faze cu ajutorul algoritmului simplex.

Faza I — *Tabelul 2.9*

	Baza	c_B	$\bar{X}$	0	0	0	0	0	0	1
				a_1	a_2	a_3	a_4	a_5	a_6	a_7
T_0	a_7	1	15	③	2	1	0	0	0	1
	a_4	0	10	1	0	0	1	0	0	0
	a_5	0	5	0	1	0	0	1	0	0
	a_6	0	3	0	0	1	0	0	1	0
	$z_j - c_j$	$W^{(0)}$	15	3	2	1	0	0	0	0
T_1	a_1	0	5	1	2/3	1/3	0	0	0	+1/3
	a_4	0	5	0	−2/3	1/3	1	0	0	−1/3
	a_5	0	5	0	1	0	0	1	0	0
	a_6	0	3	0	0	1	0	0	1	0
	$z_j - c_j$	$W^{(1)}$	0	0	0	0	0	0	0	−1

Scanned with OKEN Scanner

Se vede că s-a eliminat variabila artificială x_7 și prin urmare faza întîi s-a încheiat. Am obținut o soluție inițială de bază și putem trece la faza următoare.

Dacă nu ne interesează inversa bazei curente atunci se renunță la coloana corespunzătoare vectorului a_7. În cazul că dorim să reținem, inversa bazei, pentru nevoile unor calcule ulterioare, se menține coloana vectorului a_7 însă nu se ia în considerare diferența ΔZ_7 la aplicarea criteriului de intrare în bază (deoarece $x_7 = 0$).

Faza a II-a

Tabelul 2.10

	Baza	c_B	$\bar{X}$	1	2	3	0	0	0	1
				a_1	a_2	a_3	a_4	a_5	a_6	a_7
T_1	a_1	1	5	1	2/3	1/3	0	0	0	1/3
	a_4	0	5	0	−2/3	−1/3	1	0	0	−1/3
	a_5	0	5	0	1	0	0	1	0	0
	a_6	0	3	0	0	①	0	0	1	0
	$z_j - c_j$	$\bar{z}^{(1)} = 5$		0	−4/3	−8/3	0	0	0	—
T_2	a_1	1	4	1	2/3	0	0	0	−1/3	1/3
	a_4	0	6	0	−2/3	0	1	0	1/3	−1/3
	a_5	0	5	0	①	0	0	1	0	0
	a_3	3	3	0	0	1	0	0	1	0
	$z_j - c_j$	$\bar{z}^{(2)} = 13$		0	−4/3	0	0	0	8/3	—
T_3	a_1	1	2/3	1	0	0	0	−2/3	−1/3	1/3
	a_4	0	28/3	0	0	0	1	2/3	1/3	−1/3
	a_2	2	5	0	1	0	0	1	0	0
	a_3	3	3	0	0	1	0	0	1	0
	$z_j - c_j$	$\bar{z}^{(3)} = 53/3$		0	0	0	0	4/3	8/3	—

Observăm că toate diferențele sînt acum pozitive sau nule și deci am obținut soluția optimă a problemei de programare liniară dată.

2.7.2. Metoda bazei artificiale cu coeficienții de penalizare

Am văzut că varianta celor două faze ne dă posibilitatea să construim în mod artificial o bază de pornire pentru a putea aplica apoi algoritmul simplex.

Să presupunem că avem dată problema de programare liniară.

$$\mathbf{AX} = b \qquad \mathbf{X} \geqslant 0 \qquad \min\ (\max)\ [\mathbf{Z}(x) = \mathbf{CX}]$$

unde matricea coeficienților tehnologici $\mathbf{A}$ este de tipul $(m \times n)$ cu $m < n$ și nu avem nici o informație asupra rangului matricii $\mathbf{A}$ și deci nu putem pune în evidență o bază.

Construcţia unei baze iniţiale de pornire în varianta penalizărilor constă în următoarele :

Se înlocuieşte problema dată cu o problemă modificată sau „problema extinsă" şi anume :

$$\min\ [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X} + \Lambda'\mathbf{X}^a$$

$$\mathbf{AX} + \mathbf{IX}^a = \mathbf{b}$$

$$\mathbf{X} \geqslant 0,\ \mathbf{X}^a \geqslant 0$$

$\Lambda' = (\lambda, \lambda, \ldots, \lambda)$ este vectorul penalizărilor avînd m componente toate egale cu λ, un număr pozitiv, suficient de mare. Dacă funcţia obiectiv trebuie maximizată atunci problema extinsă capătă forma :

$$\mathbf{AX} + \mathbf{IX}^a = \mathbf{b}$$

$$\mathbf{X} \geqslant 0,\ \mathbf{X}^a \geqslant 0$$

$$\max\ [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X} - \Lambda'\mathbf{X}^a$$

cu aceleaşi notaţii pe care le-am menţionat.

Dacă problema iniţială are soluţie optimă finită, ea va fi soluţie optimă şi pentru problema extinsă.

Reciproc, orice soluţie optimă a problemei extinse, care nu conţine nici o variabilă artificială strict pozitivă, este o soluţie optimă a problemei iniţiale.

Dacă problema iniţială are funcţia obiectiv nemărginită (corespunzător categoriei ei de optim) şi problema extinsă va avea funcţia obiectiv nemărginită ; reciproca este de asemenea adevărată.

De aceea este suficient să rezolvăm problema extinsă, unde s-a pus în evidenţă o soluţie posibilă de bază, pentru a determina soluţia optimă a problemei iniţiale.

În prima etapă se rezolvă problema extinsă pentru care există baza iniţială evidentă căreia îi corespunde soluţia de bază

$$\mathbf{X}^a = \mathbf{b}$$

(adică $x_1^a = b_1 ; x_2^a = b_2, \ldots, x_m^a = b_m$).

Pentru a determina vectorul ce intră în bază se alege acea diferenţă $z_j - c_j$ care conţine termenul în λ cu cel mai mare coeficient pozitiv (dacă e o problemă de minim) sau cu cel mai mic coeficient negativ (dacă e o problemă de maxim).

Dacă un vector artificial a fost eliminat din bază putem să nu mai calculăm în continuare componentele lui în noua bază.

• Etapa întîi se încheie cînd în urma aplicării algoritmului simplex întîlnim una din situaţiile :

a) au fost eliminaţi din bază toţi vectorii artificiali.

În felul acesta s-a determinat o soluţie iniţială de bază a problemei iniţiale cu care se trece la etapa a doua cînd se rezolvă problema iniţială.

b) din bază nu s-au eliminat toţi vectorii artificiali, dar necunoscutele corespunzătoare vectorilor artificiali rămaşi în bază sînt nule şi sînt veri-

ficate condiţiile de optim. Am obţinut astfel o soluţie posibilă de bază degenerată a problemei iniţiale.

c) a rămas în bază cel puţin un vector artificial căruia îi corespunde o variabilă strict pozitivă şi sînt verificate condiţiile de negativitate (sau pozitivitate) a diferenţelor $z_j - c_j$.

În acest caz problema iniţială nu are soluţie posibilă de bază şi deci nu are soluţie optimă.

• În etapa a doua se continuă aplicarea algoritmului simplex cu soluţia de bază obţinută la prima etapă pînă ce toate cantităţile $z_j - c_j$ sînt negative (respectiv pozitive) sau se arată că problema nu admite soluţia optimă.

Exemplul 2.8. Să se determine soluţia optimă a problemei de programare liniară :

$$\max\ [\mathbf{Z}(x)] = 3x_1 + 2x_2$$
$$x_1 + 3x_2 \geqslant 9$$
$$2x_1 + x_2 \geqslant 6$$
$$-4x_1 + 3x_2 \leqslant 15$$
$$x_1 \geqslant 0,\ x_2 \geqslant 0$$

După aducerea problemei la forma standard şi adăugarea variabilelor artificiale x_6 şi x_7 se obţine următoarea problemă extinsă :

$$\max[\mathbf{Z}(x)] = 3x_1 + 2x_2 - \lambda x_6 - \lambda x_7$$
$$x_1 + 3x_2 - x_3 + x_6 = 9$$
$$2x_1 + x_2 - x_4 + x_7 = 6$$
$$-4x_1 + 3x_2 + x_5 = 15$$
$$x_j \geqslant 0\ (j = 1, 2, \ldots, 7)$$

Rezolvarea problemei este dată în tabelul 2.11.

Întrucît componentele vectorului a_3 care trebuie să intre în bază sînt toate negative, problema are optim infinit. Aceasta înseamnă că în cazul cînd $x_3 = \theta$ ($x_2 = x_6 = x_7 = 0$), atunci $x_4 = 12 + \theta$, $x_1 = 9 + \theta$, $x_5 = 15 + \theta$ şi prin urmare valoarea funcţiei $\mathbf{Z}(x) = 27 + \theta$ va creşte nelimitat (vezi ultimul tabel simplex).

Exemplul 2.9. Să se determine soluţia optimă a problemei.

$$\min[\mathbf{Z}(x)] = x_1 + 2x_2$$
$$5x_1 + 2x_2 \geqslant 10$$
$$-2x_1 + 3x_2 \leqslant 6$$
$$3x_1 + 4x_2 \leqslant 12$$
$$3x_1 \leqslant 2$$
$$x_1 \geqslant 0,\ x_2 \geqslant 0$$

Scanned with OKEN Scanner

Tabelul 2.11

B	c_B	$\bar{X}$	3	2	0	0	0	$-\lambda$	$-\lambda$	θ_i
			a_1	a_2	a_3	a_4	a_5	a_6	a_7	
a_6	$-\lambda$	9	1	③	-1	0	0	1	0	3
a_7	$-\lambda$	6	2	1	0	-1	0	0	1	6
a_5	0	15	-4	3	0	0	1	0	0	5
	$z_j - c_j$	-15λ	$-3-3\lambda$	$-2-4\lambda$	λ	λ	0	0	0	—
a_2	2	3	1/3	1	$-1/3$	0	0	1/3	0	9
a_7	$-\lambda$	3	5/3	0	1/3	-1	0	$-1/3$	1	9/5
a_5	0	6	-5	0	1	0	1	-1	0	—
	$z_j - c_j$	$6-3\lambda$	$-7/3-5/3\lambda$	0	$-2/3-1/3\lambda$	λ	0	$2/3+4/3\lambda$	0	
a_2	2	12/5	0	1	$-2/5$	(1/5)	0	2/5	$-1/5$	12
a_1	3	9/5	1	0	1/5	$-3/5$	0	$-1/5$	3/5	—
a_5	0	15	0	0	2	-3	1	-2	3	—
	$z_j - c_j$	51/5	0	0	$-1/5$	$-7/5$	0	$1/5+\lambda$	$7/5+\lambda$	—
a_4	0	12	0	5	-2	1	0	2	-1	
a_1	3	9	1	3	-1	0	0	1	0	
a_5	0	51	0	15	-4	0	1	4	0	
	$z_j - c_j$	27	0	7	-3	0	0	$3+\lambda$	λ	

Tabelele 2.12

B	c_B	$\bar{X}$	1	2	0	0	0	0	λ
			a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_7	λ	10	5	2	-1	0	0	0	1
a_4	0	6	-2	3	0	1	0	0	0
a_5	0	12	3	4	0	0	1	0	0
a_6	0	2	③	0	0	0	0	1	0
$z_j - c_j$		10λ	$-1+5\lambda$	$-2+2\lambda$	$-\lambda$	0	0	0	0
a_7	λ	10/3	0	2	-1	0	0	$-5/3$	1
a_4	0	22/3	0	③	0	1	0	2/3	0
a_5	0	10	0	4	0	0	1	-1	0
a_1	1	2/3	1	0	0	0	0	1/3	0
$z_j - c_j$		$2/3+20/3\lambda$	0	$-2+2\lambda$	0	0	0	$1/3-5/3\lambda$	0
a_7	λ	16/9	0	0	-1	$-2/3$	0	$-19/9$	1
a_2	2	22/9	0	1	0	1/3	0	2/9	0
a_5	0	2/9	0	0	0	$-4/3$	1	$-17/9$	0
a_1	1	2/3	1	0	0	0	0	1/3	0
$z_j - c_j$		$50/9+16/9\lambda$	0	0	$-\lambda$	$1/3-2/3\lambda$	0	$7/3-19/9\lambda$	0

Scanned with OKEN Scanner

După aducerea problemei la forma standard și adăugarea variabilei artificiale x_7 se obține următoarea problemă extinsă:

$$\begin{aligned}
\min\ [\mathbf{Z}(x)] = x_1 \ &+\ 2x_2 + \lambda x_7 \\
5x_1 + 2x_2 - x_3 \ &+ x_7 = 10 \\
-2x_1 + 3x_2 \ &+ x_4 \ = 6 \\
3x_1 + 4x_2 \ &+ x_5 \ = 12 \\
3x_1 \ &+ x_6 \ = 2 \\
x_j \geqslant 0 \ &(j = 1,2,\ldots,7)
\end{aligned}$$

În tabelele 2.12 este dată rezolvarea problemei de mai sus.

Deoarece $\Delta z_j \leqslant 0$ $(j = 1, 2, \ldots 7)$ și ultima bază conține variabila artificială x_7, rezultă că problema inițială nu are soluție optimă.

2.8. Problema duală

2.8.1. Concepte fundamentale

Fiecărei probleme de programare liniară definită în paragrafele precedente îi corespunde o nouă problemă de programare liniară denumită duala acesteia. Ca urmare putem vorbi de existența unui cuplu, primal-dual, de probleme de programare liniară care se pot utiliza în modelarea matematică a fenomenelor economice. Interpretarea economică a problemei duale aduce informații suplimentare pentru analiza proceselor economice și fundamentarea deciziilor.

Înțelegerea conținutului economic al parametrilor problemei duale presupune discutarea prealabilă a corespondenței care există între modelul matematic al problemei primale și modelul matematic al problemei duale.

Pentru a scrie duala unei probleme de programare liniară se atașează o variabilă u_i fiecărei restricții a problemei primale, deci în problema dual vor apărea variabilele $u_1, u_2, \ldots, u_m$.

Coeficienții funcției obiectiv a problemei primale devin termeni liberi în problema duală, iar termenii liberi din problema primală devin coeficienți ai funcției obiectiv a problemei duale. Prin trecere la dual, o problemă de maxim se transformă în problemă de minim și invers.

Fiecărei variabile x_j din problema primală îi corespunde o restricție în problema duală, prin urmare coeficienții variabilei x_j devin coeficienți ai unei restricții în problema duală. În acest fel problema duală are n restricții iar coeficienții acestora formează matricea $\mathbf{A}'$ (transpusa matricei $\mathbf{A}$ din problema primală). Trebuie subliniat faptul că variabilelor nenegative din problema primală le corespund inegalități concordante în problema dual. Această precizare este deosebit de importantă pentru aplicațiile

Scanned with OKEN Scanner

programării liniare în economie întrucît modelele folosite conţin variabile nenegative*.

Domeniul de variaţie al variabilelor u_i $(i = 1, 2, \ldots, m)$ se stabileşte pe baza corespondenţei dintre restricţia $\mathbf{R}_i$ din problema primală şi variabila duală u_i ce i se ataşează. Astfel, restricţiilor concordante le corespund variabile $u_i \geqslant 0$, celor neconcordante le corespund variabile $u_i \leqslant 0$, iar restricţiilor scrise sub formă de egalităţi le corespund variabile duale ce nu sînt supuse restricţiilor de semn.

Din cele arătate mai sus rezultă că între problema primală şi duală există o corespondenţă biunivocă care este prezentată sintetic în tabelul 2.13 :

Tabelul 2.13

	Problema primală	Problema duală
I	$\max\ [Z(x)] = \sum_{j=1}^{n} c_j x_j$ în condiţiile $\sum_{j=1}^{n} a_{hj} x_j \leqslant b_h \quad h \in I_1$ $\sum_{j=1}^{n} a_{kj} x_j \geqslant b_k \quad k \in I_2$ $\sum_{j=1}^{n} a_{lj} x_j = b_l \quad l \in I_3$ $x_j \geqslant 0 \quad (j = 1,2,\ldots n)$	$\min\ [G(u)] = \sum_{i=1}^{m} b_i u_i$ în condiţiile $\sum_{i=1}^{m} a_{ij} u_i \geqslant c_j \quad (j = 1, 2, \ldots, n)$ $u_h \geqslant 0 \qquad h \in I_1$ $u_k \leqslant 0 \qquad k \in I_2$ u_l oarecare $l \in I_3$
II.	$\min\ [Z(x)] = \sum_{j=1}^{n} c_j x_j$ în condiţiile $\sum_{j=1}^{n} a_{hj} x_j \leqslant b_h \quad h \in \mathbf{I}_1$ $\sum_{j=1}^{n} a_{kj} x_j \geqslant b_k \quad k \in \mathbf{I}_2$ $\sum_{j=1}^{n} a_{lj} x_j = b_l \quad l \in \mathbf{I}_3$	$\max\ [G(u)] = \sum_{i=1}^{m} b_i u_i$ în condiţiile $\sum_{i=1}^{m} a_{ij} u_i \leqslant c_i \quad (j = 1,2,\ldots,n)$ $u_h \leqslant 0 \quad h \in \mathbf{I}_1$ $u_k \geqslant 0 \quad k \in \mathbf{I}_2$ u_l oarecare $l \in \mathbf{I}_3$

Observaţie : Pentru simplificare s-au scris doar relaţiile corespunzătoare celor trei tipuri de restricţii ce pot apărea într-un model de programare liniară. Precizăm că $\mathbf{I}_1 \cup \mathbf{I}_2 \cup \mathbf{I}_3 = (1, 2, \ldots m.)$

* Dacă variabilele problemei primale sînt nepozitive, restricţiile corespunzătoare din problema duală sînt scrise sub formă de inegalităţi neconcordante. Unor variabile fără restricţii de semn în problema primală le corespund în problema duală restricţii sub formă de egalităţi. Aplicînd aceste reguli problemei duale se obţine modelul matematic al problemei primale.

Scanned with OKEN Scanner

Corespondența biunivocă a cuplurilor de probleme prezentate în tabelul 2.13 este pusă în evidență de teorema dualității.

Teoremă. Într-un cuplu de probleme primală-duală poate apărea una dintre situațiile :

— una dintre probleme are soluție optimă finită și atunci cealaltă problemă are soluție, iar valorile externe ale funcțiilor obiectiv sînt egale ;

— una dintre probleme are optim infinit și cealaltă problemă nu are soluție ;

— nici una dintre probleme nu are soluție.

În continuare vom da demonstrația teoremei dualității pentru cuplul :

$$\max [\mathbf{Z}(x)] = \mathbf{C'X} \qquad \min [\mathbf{G}(u)] = \mathbf{b'U}$$

$$\mathbf{AX} \leqslant \mathbf{b} \qquad \mathbf{A'U} \geqslant \mathbf{C}$$

$$\mathbf{X} \geqslant 0 \qquad \mathbf{U} \geqslant 0$$

care rezultă din forma canonică a problemei I(vezi tabelul 2.13).

Pentru a demonstra prima afirmație a teoremei dualității în cazul cuplului scris mai sus, trebuie să arătăm că :

a) dacă problema primală admite o soluție finită, problema duală admite și ea o soluție ;

b) maximul funcției de eficiență a problemei primale este egal cu minimul funcției de eficiență a problemei duale.

Considerăm că problema este nedegenerată și are soluție optimă, deci problema are o soluție de bază optimă care cuprinde exact m componente nenule ale variabilelor. Pentru simplificare vom admite că primele m variabile au valori nenule în soluția de bază optimă a problemei primale, ca urmare variabilele nenule din soluția optimă satisfac sistemul :

$$\begin{aligned} &a_{11}x_1^0 + a_{12}x_2^0 + \ldots + a_{1m}x_m^0 = b_1 \\ &a_{21}x_1^0 + a_{22}x_2^0 + \ldots + a_{2m}x_m^0 = b_2 \\ &\cdots\cdots\cdots\cdots\cdots\cdots \\ &a_{m1}x_1^0 + a_{m2}x_2^0 + \ldots + a_{mm}x_m^0 = b_m \end{aligned} \tag{2.13}$$

Dacă notăm cu $\mathbf{B}$ matricea formată din componentele vectorilor a_1, a_2, ..., a_m, sistemul (2.13) se scrie astfel :

$$\mathbf{BX^0} = \mathbf{b}$$

Condiția de optimalitate a problemei analizate este dată de relația :

$$\Delta z = \mathbf{C'_B B^{-1} \cdot A} - \mathbf{C'} \geqslant 0 \tag{2.14}$$

unde $\mathbf{B^{-1} \cdot A}$ reprezintă matricea coeficienților tabelului simplex final, iar $\mathbf{C'_B}$ vectorul linie al coeficienților variabilelor care formează baza optimă.

Scanned with OKEN Scanner

Vom demonstra că și problema duală are o soluție corespunzătoare bazei **B**. În acest scop presupunem că vectorul $\mathbf{U}_0' = (u_1^0, u_2^0, \ldots, u_m^0)$ se determină din relația :

$$\mathbf{U}_0' = \mathbf{C}_B' \cdot \mathbf{B}^{-1} \tag{2.15}$$

Introducînd (2.15) în (2.14) obținem $\mathbf{U}_0'\mathbf{A} - \mathbf{C}' \geqslant 0$, de unde rezultă că :

$$\mathbf{U}_0'\mathbf{A} \geqslant \mathbf{C}'$$

sau $\mathbf{A}'\mathbf{U}_B \geqslant \mathbf{C}$

Întrucît vectorul $\mathbf{U}_0$ satisface condițiile problemei duale rezultă că el reprezintă o soluție a acesteia.

O bază **B** a problemei primale se numește dual realizabilă dacă vectorul $\mathbf{U}_B' = \mathbf{C}_B\mathbf{B}^{-1}$ este o soluție a problemei duale, deci dacă $\mathbf{A}'\mathbf{U}_B \geqslant \mathbf{C}$.

Rămîne să se demonstreze că $\mathbf{U}_0$ este soluția optimă a problemei duale. Pentru aceasta vom considera orice vector **U** din spațiul m-dimensional, care satisface restricțiile $\mathbf{A}'\mathbf{U} \geqslant \mathbf{C}$ și $\mathbf{U} \geqslant 0$, precum și orice vector **X** din spațiul n-dimensional care satisface restricțiile $\mathbf{AX} \leqslant \mathbf{b}$ și $\mathbf{X} \geqslant 0$.

Prin înmulțirea la dreapta a sistemului $\mathbf{U}'\mathbf{A} \geqslant \mathbf{C}'$ cu vectorul coloană **X** obținem :

$$\mathbf{U}'\mathbf{AX} \geqslant \mathbf{C}'\mathbf{X} = \mathbf{Z}(x) \tag{2.16}$$

Este evident că sistemul de inegalități (2.16) este satisfăcut și de $\mathbf{AX} = \mathbf{b}$, deoarece **b** este limita maximă, deci :

$$\mathbf{U}'\mathbf{b} \geqslant \mathbf{C}'\mathbf{X}$$

Înlocuind vectorul **X** cu vectorul soluției optime $\mathbf{X}^0$ a problemei inițiale rezultă :

$$\mathbf{U}'\mathbf{b} \geqslant \mathbf{C}_B' \cdot \mathbf{X}^0 = \max\ [\mathbf{Z}(x)] \tag{2.17}$$

ceea ce înseamnă că valoarea funcției obiectiv a problemei duale $[\mathbf{G}(u) = \mathbf{U}'\mathbf{b}]$ corespunzătoare oricărei soluții a sistemului său de inecuații este cel puțin egală cu valoarea maximă a funcției obiectiv a problemei primale.

Dacă în (2.17) înlocuim vectorul $\mathbf{U}'$ cu $\mathbf{U}_0'$ ajungem la expresia

$$\mathbf{U}_0'\mathbf{b} = \mathbf{C}\mathbf{B}'\mathbf{B}^{-1} \cdot \mathbf{b} \geqslant \mathbf{C}_B'\,\mathbf{X}^0 = \max\ [\mathbf{Z}(x)]$$

de unde, avînd în vedere că $\mathbf{B}^{-1}\mathbf{b} = \mathbf{X}^0$, rezultă :

$$\mathbf{U}_0' \cdot \mathbf{b} = \mathbf{C}_B'\,\mathbf{X}^0 \tag{2.18}$$

Prin urmare $\mathbf{U}_0'$ este soluția optimă a problemei duale, iar valoarea funcției obiectiv corespunzătoare acestei soluții este dată de relația (2.18), deci se poate scrie :

$$\min\ [\mathbf{G}(u)] = \max\ [\mathbf{Z}(x)]$$

Se poate demonstra cu ușurință că dacă funcția obiectiv a problemei primale este nelimitată, adică $\max\ [\mathbf{Z}(x)] = \infty$, atunci problema duală nu are soluție. În cazul examinat relația (2.17) devine : $\mathbf{G}(u) \geqslant \infty$.

Scanned with OKEN Scanner

Aceasta înseamnă că orice soluție $\mathbf{U}$ a sistemului $\mathbf{A'U} \geqslant \mathbf{C}$ trebuie să corespundă unei valori a funcției obiectiv $G(w)$ cel puțin egală cu ∞. Întrucît relația de mai sus nu are sens rezultă că problema duală nu are soluție.

Din cele arătate mai sus se desprinde concluzia că tabelul simplex final corespunzător problemei primale conține atît soluția optimă a problemei primale cît și a problemei duale. Soluția problemei primale $(\mathbf{X}^0)$ se obține pe coloana vectorului a_0, iar soluția problemei duale $\mathbf{U}_0 = \mathbf{C}'_B\mathbf{B}^{-1}$ se obține pe linia z la intersecția cu coloanele vectorilor care au format baza inițială.

Teorema ecarturilor complementare

O condiție necesară și suficientă pentru ca un cuplu de soluții admisibile de bază $\mathbf{X}^0$ și $\mathbf{U}^0$ să dea optim, este ca soluțiile să verifice simultan relațiile

$$\left[c_j - \sum_{i=1}^{m} u_i^0 \cdot a_{ij}\right] x_j^0 = 0 \qquad (j = 1, 2, \ldots, n) \tag{2.16}$$

$$\left[\sum_{j=1}^{n} a_{ij} x_j^0 - b_i\right] u_i^0 = 0 \qquad (i = 1, 2, \ldots, m)$$

Acestea se scriu sub formă matricenlă, astfel:

$$(\mathbf{C}'_0 - \mathbf{U}'_0\mathbf{A})\mathbf{X}^0 = 0$$

$$\mathbf{U}'_0(\mathbf{AX}^0 - \mathbf{b}) = 0$$

Pe baza acestor relații se pot formula următoarele concluzii:

a) dacă $u_i^0 > 0$ atunci $\sum_{j=1}^{n} a_{ij} x_j^0 = b_i$;

b) dacă $\sum_{j=1}^{n} a_{ij} x_j^0 < b_i$ atunci $u_i^0 = 0$;

c) dacă $x_j^0 > 0$ atunci $\sum_{i=1}^{m} u_i a_{ij} = c_j$;

d) dacă $\sum_{i=1}^{m} u_i a_{ij} > c_j$ atunci $x_j = 0$.

Prima relație arată că variabila duală corespunzătoare unei resurse utilizată în întregime are o valoare pozitivă, iar cea de a doua arată că $u_i^0 = 0$ dacă resursa i nu este utilizată în întregime.

Analizînd punctele c și d se trage concluzia că dacă costul unitar al activității a_{ij} obținut prin evaluarea consumurilor specifice a_{ij} cu ajutorul componentelor soluției optime a problemei duale, este egal cu coeficientul c_j din funcția obiectiv (de exemplu beneficiul unitar) atunci componenta

Aceasta înseamnă că orice soluție $\mathbf{U}$ a sistemului $\mathbf{A'U} \geqslant \mathbf{C}$ trebuie să corespundă unei valori a funcției obiectiv $G(u)$ cel puțin egală cu c_0. Întrucît relația de mai sus nu are sens rezultă că problema duală nu are soluție.

Din cele arătate mai sus se desprinde concluzia că tabelul simplex final corespunzător problemei primale conține atît soluția optimă a problemei primale cît și a problemei duale. Soluția problemei primale ($\mathbf{X}^0$) se obține pe coloana vectorului a_0, iar soluția problemei duale $\mathbf{U}_0 = \mathbf{C}'_B\mathbf{B}^{-1}$ se obține pe linia z la intersecția cu coloanele vectorilor care au format baza inițială.

Teorema ecarturilor complementare

O condiție necesară și suficientă pentru ca un cuplu de soluții admisibile de bază $\mathbf{X}^0$ și $\mathbf{U}$ să dea optim, este ca soluțiile să verifice simultan relațiile

$$\left[c_j - \sum_{i=1}^{m} u_i^0 \cdot a_{ij}\right] x_j^0 = 0 \qquad (j = 1, 2, \ldots, n) \tag{2.16}$$

$$u_i^0 \left[\sum_{j=1}^{n} a_{ij} x_j^0 - b_i\right] = 0 \qquad (i = 1, 2, \ldots, m)$$

Acestea se scriu sub formă matricelă astfel:

$$\mathbf{U}'_0(\mathbf{AX}^0 - \mathbf{b}) = 0$$

$$(\mathbf{C}'_0 - \mathbf{U}'_0\mathbf{A})\mathbf{X}^0 = 0$$

Pe baza acestor relații se pot formula următoarele concluzii:

a) dacă $u_i^0 > 0$ atunci $\sum_{j=1}^{n} a_{ij}x_j^0 = b_i$;

b) dacă $\sum_{j=1}^{n} a_{ij}x_j^0 < b_i$ atunci $u_i^0 = 0$;

c) dacă $x_j^0 > 0$ atunci $\sum_{i=1}^{m} u_i a_{ij} = c_j$;

d) dacă $\sum_{i=1}^{m} u_i a_{ij} > c_j$ atunci $x_j = 0$.

Prima relație arată că variabila duală corespunzătoare unei resurse utilizată în întregime are o valoare pozitivă, iar cea de a doua arată că $u_i^0 = 0$ dacă resursa i nu este utilizată în întregime.

Analizînd punctele c și d se trage concluzia că dacă costul unitar al activității a_j, obținut prin evaluarea consumurilor specifice a_{ij} cu ajutorul componentelor soluției optime a problemei duale, este egal cu coeficientul c_j din funcția obiectiv (de exemplu beneficiul unitar) atunci componenta

$x_j^0 > 0$, iar dacă acest cost este mai mare decît c_j, atunci nu este eficient să includem în programul optim activitatea j (ca urmare $x_j^0 = 0$).

Exemple numerice

În continuare vom ilustra tehnica de construire a problemei duale pe baza cîtorva tipuri de probleme de programare liniară.

Exemplul 2.10. Considerăm o unitate economică care fabrică produsele P_1, P_2 și P_3. Pentru obținerea lor se utilizează trei resurse : forța de muncă, mijloace de muncă și materii prime. În tabelul următor se dau consumurile specifice și cantitățile disponibile din cele trei resurse, precum și prețurile de vînzare ale produselor.

Tabelul 2.14

Resurse \ Produse	P_1	P_2	P_3	Disponibil (unități fizice)
Forța de muncă	1	3	4	15
Mijloace de muncă	2	5	1	10
Materii prime	4	1	2	25
Preț de vînzare (unități monetare)	3	2	6	—

Modelul matematic pe baza căruia se stabilește programul optim de producție, avînd drept criteriu de eficiență valoarea maximă a producției, are forma :

$$\max\ [\mathbf{Z}\,(x)] = 3x_1 + 2x_2 + 6x_3$$

în condițiile :

$$\begin{aligned} x_1 + 3x_2 + 4x_3 &\leqslant 15 \\ 2x_1 + 5x_2 + x_3 &\leqslant 10 \\ 4x_1 + x_2 + 2x_3 &\leqslant 25 \\ x_j &\geqslant 0 \quad (j = 1,2,3) \end{aligned} \tag{2.19}$$

Utilizînd regulile de dualitate prezentate mai înainte rezultă următoarea problemă duală de programare liniară :

$$\min\ [\mathbf{G}\,(u)] = 15u_1 + 10u_2 + 25u_3$$

în condițiile

$$\begin{aligned} u_1 + 2u_2 + 4u_3 &\geqslant 3 \\ 3u_1 + 5u_2 + u_3 &\geqslant 2 \\ 4u_1 + u_2 + 2u_3 &\geqslant 6 \\ u_j &\geqslant 0 \quad (j = 1,2,3) \end{aligned}$$

Scanned with OKEN Scanner

$z_j^0 > 0$, iar dacă acest cost este mai mare decît c_j, atunci nu este eficient să includem în programul optim activitatea j (ca urmare $x_j^0 = 0$).

Exemple numerice

În continuarea vom ilustra tehnica de construire a problemei duale pe baza citorva tipuri de probleme de programare liniară.

Exemplul 2.10. Considerăm o unitate economică care fabrică produsele P_1, P_2 și P_3. Pentru obținerea lor se utilizează trei resurse : forța de muncă, mijloace de muncă și materii prime. În tabelul următor se dau consumurile specifice și cantitățile disponibile din cele trei resurse, precum și prețurile de vînzare ale produselor.

Tabelul 2.14

Resurse \ Produse	P_1	P_2	P_3	Disponibil (unități fizice)
Forța de muncă	1	3	4	15
Mijloace de muncă	2	5	1	10
Materii prime	4	1	2	25
Preț de vînzare (unități monetare)	3	2	6	—

Modelul matematic pe baza căruia se stabilește programul optim de producție, avînd drept criteriu de eficiență valoarea maximă a producției, are forma :

$$\max\ [\mathbf{Z}\,(x)] = 3x_1 + 2x_2 + 6x_3$$

în condițiile :

$$\begin{aligned} x_1 + 3x_2 + 4x_3 &\leqslant 15 \\ 2x_1 + 5x_2 + x_3 &\leqslant 10 \\ 4x_1 + x_2 + 2x_3 &\leqslant 25 \\ x_j &\geqslant 0 \quad (j = 1,2,3) \end{aligned} \tag{2.19}$$

Utilizînd regulile de dualitate prezentate mai înainte rezultă următoarea problemă duală de programare liniară :

$$\min\ [\mathbf{G}\,(u)] = 15u_1 + 10u_2 + 25u_3$$

în condițiile

$$\begin{aligned} u_1 + 2u_2 + 4u_3 &\geqslant 3 \\ 3u_1 + 5u_2 + u_3 &\geqslant 2 \\ 4u_1 + u_2 + 2u_3 &\geqslant 6 \\ u_j &\geqslant 0 \quad (j = 1,2,3) \end{aligned}$$

Scanned with OKEN Scanner

Soluția optimă a problemei primale este prezentată în tabelul simplex final (tabelul 2.15).

După cum s-a arătat, prin rezolvarea uneia dintre problemele cuplului primală — duală se obțin soluțiile ambelor probleme. În cazul examinat, soluția optimă a problemei duale se citește pe linia z la intersecția cu coloanele vectorilor a_4, a_5 și a_6 care au format baza inițială, deci :

$$u_1^0 = \frac{9}{7},\ u_2^0 = \frac{6}{7} \text{ și } u_3^0 = 0.$$

Tabelul 2.15

B	C_i	P_0	3	2	6	0	0	0
			a_1	a_2	a_3	a_4	a_5	a_6
a_3	6	20/7	0	1/7	1	2/7	−1/7	0
a_1	3	25/7	1	17/7	0	−1/7	4/7	0
a_6	0	35/7	0	−9	0	0	−2	1
z_j	—	195/7	3	57/7	6	9/7	6/7	0
ΔZ_j	—	—	0	43/7	0	9/7	6/7	0

Este ușor de verificat că $\mathbf{U}_0' = \mathbf{C}_B' \cdot \mathbf{B}^{-1}$. Spre exemplu u_1^0 rezultă din relația :

$$u_1^0 = \sum_{k=1}^{3} c_k b_{k1} = 6 \cdot \frac{2}{7} - 3 \cdot \frac{1}{7} = -\frac{9}{7}$$

Pe baza rezultatelor obținute se verifică imediat ca $\max\ [\mathbf{Z}(x)] = \min\ [\mathbf{G}(u)]$:

$$\max\ [\mathbf{Z}(x)] = 3 \cdot \frac{25}{7} + 6 \cdot \frac{20}{7} = \frac{195}{7} \text{ unități monetare}$$

$$\min\ [\mathbf{G}(u)] = 15 \cdot \frac{9}{7} + 10 \cdot \frac{6}{7} = \frac{195}{7} \text{ unități monetare.}$$

Exemplul 2.11. Activitatea unei întreprinderi industriale se concretizează în fabricarea produselor omogene P_1, P_2, P_3 și P_4 ale căror costuri unitare de producție sînt : $c_1 = 4$, $c_2 = 3$, $c_3 = 7$ respectiv $c_4 = 2$ unități monetare.

Pentru desfășurarea procesului de producție este necesar ca din produsul P_1 să se asigure un stoc de cel puțin 5 unități și cîte cel puțin 2 unități pentru fiecare piesă P_3 (produsul P_1 intră în componența produsului P_3).

În anul de bază volumul producției realizate de întreprindere a fost de 20 unități. În urma analizei desfacerilor s-a ajuns la concluzia că cererea este în creștere.

Scanned with OKEN Scanner

De asemenea, întreprinderea își propune ca în perioada de plan să obțină un beneficiu total cel puțin egal cu cel din perioada de bază care a fost de *100* u.m. Beneficiul unitar corespunzător celor patru produse s-a estimat a fi de *2*, *5*, *8* și respectiv *3* u.m.

Modelul matematic corespunzător problemei primale și respectiv problemei duale are forma :

$$\min [\mathbf{Z}(x)] = 4x_1 + 3x_2 + 7x_3 + 2x_4 \qquad \max [\mathbf{G}(u)] = 5u_1 + 20u_2 + 100u_3$$

în condițiile

$$x_1 - 2x_3 \geqslant 5$$
$$x_1 + x_2 + x_3 + x_4 \geqslant 20$$
$$2x_1 + 5x_2 + 8x_3 + 3x_4 \geqslant 100$$
$$x_j \geqslant 0 \quad (j = 1,2,3,4)$$

în condițiile

$$u_1 + u_2 + 2u_3 \leqslant 4$$
$$u_2 + 5u_3 \leqslant 3$$
$$-2u_1 + u_2 + 8u_3 \leqslant 7$$
$$u_2 + 3u_3 \leqslant 2$$
$$u_j \geqslant 0 \quad (j = 1,2,3)$$

Exemplul 2.12. Considerăm problema dată în exemplul *2.10* căreia îi atașăm două modificări :

a) forța de muncă trebuie utilizată în întregime ;

b) volumul producției planificate al produsului P_2 trebuie să fie cel puțin egal cu *1* unități fizice.

Cuplul de probleme, primală-duală, care exprimă matematic problema economică descrisă, are forma :

$$\max [\mathbf{Z}(x)] = 3x_1 + 2x_2 + 6x_3 \qquad \min [\mathbf{G}(u)] = 15u_1 + 10u_2 + 25u_3 + u_4$$

în condițiile

$$x_1 + 3x_2 + 4x_3 = 15$$
$$2x_1 + 5x_2 + x_3 \leqslant 10$$
$$4x_1 + x_2 + 2x_3 \leqslant 25$$
$$x_2 \geqslant 1$$
$$x_j \geqslant 0 \quad (j = 1,2,3)$$

în condițiile

$$u_1 + 2u_2 + 4u_3 \geqslant 3$$
$$3u_1 + 5u_2 + u_3 + u_4 \geqslant 2$$
$$4u_1 + u_2 + 2u_3 \geqslant 6$$
$$u_1 \text{ oarecare}$$
$$u_2 \geqslant 0$$
$$u_3 \geqslant 0$$
$$u_4 \leqslant 0$$

Analizînd modelele de mai sus se constată că variabila u_1 nu este supusă reacției de semn, deoarece în problema primală prima restricție este o ecuație. De asemenea se observă că restricțiilor a doua și a treia din problema primală, care sînt concordante, le corespund variabile duale pozitive, iar restricției a patra care este neconcordanta îi corespunde o variabilă nepozitivă ($u_4 \leqslant 0$).

2.8.2. Interpretarea economică a problemei duale

În paragrafele precedente s-a demonstrat că prin rezolvarea unei probleme de programare liniară se obține atît soluția optimă a problemei inițiale cît și a problemei duale. Corespondența dintre problema inițială

Scanned with OKEN Scanner

și dual ă se reflectă și în conținutul economic al parametrilor incluși în soluția problemei duale. Semnificația economică a indicatorilor u_i este determinată de natura problemei economice și de tipul restricțiilor problemei primale.

Într-o problemă sub formă canonică în care se cere maximizarea funcției obiectiv restricțiile pot fi interpretate ca inecuații ce se referă la resurse și de aceea interpretarea variabilelor u_i este mai simplă. Din acest motiv vom analiza în primul rînd modelul (2.19) de programare liniară. Se constată că în funcția obiectiv a problemei duale apar cantitățile disponibile din cele trei resurse înmulțite cu mărimile u_1, u_2 și respectiv u_3. Întrucît resursele reprezintă valori de întrebuințare diferite, cantitățile b_i nu se pot însuma decît dacă indicatorii u_i evaluează resursele în aceeași unitate de măsură. După cum se vede din acest model, valoarea indicatorilor u_i depinde de cantitatea de resurse disponibile, de structura consumurilor directe din resursa respectivă și de structura matematică a modelului.

Conform teoremei ecarturilor complementare, resurselor utilizate în întregime le corespund evaluări u_i pozitive, iar resurselor excedentare le corespund indicatori $u_i = 0$. Un produs P_j va apărea în soluția optimă ($x_j^0 > 0$) numai atunci cînd costul său unitar de producție, obținut prin evaluarea resurselor consumate cu ajutorul indicatorilor u_i, nu depășește prețul unitar c_j, adică cînd :

$$\sum_{i=1}^{m} u_i a_{ij} = c_j \qquad (2.20)$$

Ideea enunțată mai sus se confirmă și în cazul exemplului analizat. Astfel, produsul P_2 nu apare în soluția optimă ($x_2^0 = 0$) deoarece costul său de producție depășește prețul unitar de realizare :

$$a_{12}u_1 + a_{22}u_2 + a_{32}u_3 > c_2$$

$$\text{adică } 3 \cdot \frac{9}{7} + 5 \cdot \frac{6}{7} + 1 \cdot 0 = 8{,}1 > 2.$$

Celelalte două produse intră în programul optim întrucît este satisfăcută relația (2.20). Deci, fabricarea unui produs care nu figurează în programul optim este neeficientă din punctul de vedere al folosirii resurselor.

Prin urmare, pentru o problemă de programare liniară scrisă sub formă canonică în care se urmărește maximizarea funcției $\mathbf{Z}(x)$, mărimea u_i reprezintă valoarea ultimei unități folosite în producție din resursa $\mathbf{R}_i$. Avînd în vedere că $\max[\mathbf{Z}(x)] = \min[\mathbf{G}(u)]$, rezultă că dacă cantitatea disponibilă din resursa $\mathbf{R}_i$ crește cu o unitate, atunci valoarea funcției obiectiv crește cu u_i, deci u_i măsoară creșterea valorii funcției obiectiv determinată de creșterea cu o unitate a cantității disponibile b_i.

Aceste evaluări obținute dintr-un program optim au fost denumite în literatura de specialitate „prețuri umbră" sau „evaluări obiectiv determinate".

Scanned with OKEN Scanner

Pentru o problemă scrisă sub formă canonică, în care se cere minimizarea funcţiei $\mathbf{Z}(x)$, restricţiile se referă la caracteristici economice cărora li se impun limite inferioare sau la indicatori calitativi cu limită inferioară stabilită. În acest caz preţul umbră u_i măsoară creşterea costului total al producţiei, a cheltuielilor de muncă, a cheltuielilor cu fondurile fixe etc. determinată de creşterea cu o unitate a componentei b_i a vectorului termenilor liberi. Astfel, dacă la exemplul 2.11 planul de producţie prevede o creştere a volumului producţiei de la *20* unităţi la *21* unităţi, atunci costul total al producţiei va creşte cu u_2 unităţi.

În cazul problemelor de programare liniară care conţin restricţii neconcordante şi egalităţi, la interpretarea preţurilor umbră, trebuie să se ţină seama de natura economică a funcţiei obiectiv şi de semnul variabilei u_i. Pentru a elucida acest aspect ne vom referi la modelul matematic ce rezultă din exemplul 2.12.

Soluţiile optimale ale problemei primale şi duale sînt :

$$\mathbf{X}_0' = (x_1, x_2, x_3, x_4, x_5, x_6) = (8/7,\ 1,\ 19/7,\ 0,\ 14,\ 0)$$

$$\mathbf{U}_0' = (u_1, u_2, u_3, u_4) = (9/7,\ 6/7,\ 0, -43/7).$$

Se constată imediat că valoarea funcţiei obiectiv $\mathbf{Z}(\mathbf{X}_0) = \mathbf{G}(u_0) = 152/7$ este mai mică decît cea obţinută prin rezolvarea modelului din exemplul 2.10 cu $\Delta\mathbf{Z}(x) = 195/7 - 152/7 = 43/7$ unităţi monetare*. Această diferenţă apare datorită introducerii restricţiei $x_2 \geqslant 1$ al cărei preţ umbră este $u_4 = -43/7$. Prin urmare variabilele u_4 arată cu cît scade valoarea funcţiei obiectiv $\mathbf{Z}(x)$ atunci cînd nivelul minim impus variabilei x_2 este mai mare cu o unitate.

În sfîrşit variabila $u_1 = 9/7$, care este ataşată unei restricţii sub formă de egalitate, arată că sporirea cu o unitate a nivelului forţei de muncă disponibile conduce la creşterea valorii producţiei cu $9/7$ unităţi monetare.

Rezultă că preţurile umbră aduc informaţii suplimentare pentru analiza eficienţei economice a resurselor şi a diferiţilor indicatori economici sau tehnici care apar în restricţiile unei probleme de programare liniară. Pe baza lor se pot fundamenta deciziile privind alocarea judicioasă a resurselor, se pot stabili măsuri de stimulare a consumului raţional al resurselor, se determină cît mai corect nivelul minim şi maxim al diferiţilor indicatori tehnici şi economici de care depinde structura planului optim.

Soluţia optimă obţinută prin utilizarea datelor iniţiale poate constitui un punct de plecare pentru analiza economică privind alocarea eficientă a resurselor. Vom ilustra acest aspect folosind modelul matematic elaborat în cadrul exemplului 2.10.

În tabelul 2.16 se dau soluţiile optime ale problemei primale şi a problemei duale pentru diferite valori ale primei resurse ($b_2 = 10$, $b_3 = 25$).

Se observă că între anumite limite de variaţie ale primei resurse, preţurile umbră au aceeaşi valoare. Pentru valori mai mari decît *40* unităţi, preţurile umbră au o altă valoare, deoarece, în acest caz prima resursă şi

---

* Reamintim că exemplul 2.12 s-a obţinut din exemplul 2.10 ca urmare a unor transformări. Precizăm că în soluţiile optime ale ambelor modele prima restricţie se verifică ca egalitate. De aceea diferenţa dintre cele două soluţii este determinată numai de restricţia $x_2 \geqslant 1$.

Scanned with OKEN Scanner

a treia devin excedentare. Valoarea funcţiei de eficienţă va creşte cu $u_1 \Delta b_1$, unde Δb_1 reprezintă sporul disponibilului din prima resursă. Pentru $b_1 > 40$ preţul umbră al resursei a doua este $u_2 = 6$, deci valoarea funcţiei de eficienţă va creşte cu 6 unităţi atunci cînd b_2 creşte cu o unitate. Aceste informaţii sînt utile pentru fundamentarea planului de producţie şi aprovizionare cu diverse resurse.

Tabelul 2.16

Variabile	Cantitatea disponibilă din prima resursă (b_1)					
	15	16	26	36	40	41
u_4 / x_1	3 / 25/7	3 / 24/7	3 / 14/7	3 / 4/7	3 / 0	3 / 0
u_5 / x_2	57/7 / 0	57/7 / 0	57/7 / 0	57/7 / 0	57/7 / 0	30 / 0
u_6 / x_3	6 / 20/7	6 / 22/7	6 / 42/7	6 / 62/7	6 / 70/7	6 / 70/7
u_1 / x_4	9/7 / 0	9/7 / 0	9/7 / 0	9/7 / 0	9/7 / 0	0 / 1
u_2 / x_5	6/7 / 0	6/7 / 0	6/7 / 0	6/7 / 0	6/7 / 0	6 / 0
u_3 / x_6	0 / 5	0 / 5	0 / 5	0 / 5	0 / 5	0 / 5
$Z(X)$	195/7	204/7	294/7	384/7	420/7	60

La interpretarea influenţei pe care variaţia cantităţii disponibile b_i o are asupra preţurilor umbră, trebuie să se ţină seama că b_i este o variabilă continuă, din această cauză u_i se defineşte astfel :*

$$u_i = \frac{\partial F}{\partial b_i}$$

Este clar că u_i va avea o altă valoare cînd b_i creşte (scade) peste o anumită limită (de exemplu $b_i > 40$).

După cum s-a arătat mai înainte în cadrul algoritmului simplex se stabileşte activitatea cea mai eficientă a_k prin folosirea criteriului de intrare.

$$\Delta Z_k = \min (z_j - c_j) \text{ la problemele de maxim}$$

şi

$$\Delta Z_k = \max (z_j - c_j) \text{ la problemele de minim}$$

* F este o funcţie de tip Lagrange, obţinută din problema standard de programare liniară, astfel :

$$F = Z(x) + u_i R_i(x) \quad (i = 1, 2, \ldots, m)$$

unde $R_i(x)$ reprezintă restricţiile problemei de programare liniară, iar u_i, $1 < i < m$, multiplicatorii lui Lagrange.

Scanned with OKEN Scanner

Avînd în vedere relaţia (2.14) şi (2.15) rezultă că :

$$z_j = \sum_{i=1}^{m} c_i y_{ij} = \sum_{i=1}^{m} u_i a_{ij} \quad (j \in \mathbf{J_S})$$

unde y_{ij} reprezintă elementele coloanei j din tabelul simplex corespunzător unei soluţii de bază. În cea de a doua sumă a relaţiei precedente se evaluează coeficienţii consumurilor directe corespunzători activităţii a_j prin preţurile umbră ataşate celor m restricţii şi de aceea mărimea rezultată poate fi interpretată ca un preţ umbră al produsului sau activităţii j. Trebuie precizat că pentru fiecare bază vom dispune de un vector $\mathbf{U} = (u_1, u_2, \ldots, u_m)$ care se referă la fluxurile intrări-ieşiri corespunzătoare soluţiei de bază. De aceea evaluările privind eficienţa economică a activităţilor a_i ($j \in \mathbf{J_S}$) făcute cu ajutorul indicatorilor u_i sînt valabile numai pentru baza considerată.

Concluziile obţinute pe baza analizei corespondenţei biunivoce dintre problema primală şi cea duală ne permit să înţelegem sensul economic al criteriului de intrare în bază.

Pentru problemele în care se cere maximizarea funcţiei $\mathbf{Z}(x)$, indicatorii z_j reprezintă costul unitar de fabricaţie al produsului j rezultat prin evaluarea coeficienţilor a_{ij} prin preţurile umbră ataşate restricţiilor. Rezultă că prin calcularea diferenţei Δz_j se compară acest cost unitar cu coeficientul c_j (preţ unitar, beneficiu unitar etc.) din funcţia obiectiv. Dacă există diferenţe negative ($\Delta z_j < 0$) înseamnă că la activităţile respective preţul umbră al produsului j (costul unitar de fabricaţie exprimat în indicatori u_i) este mai mic decît venitul realizat şi de aceea activitatea a_j este eficientă. Aşa se explică că va intra în bază vectorul a_k care corespunde diferenţei Δz_j minime. În momentul în care toate diferenţele sînt pozitive ($\Delta z_j \geqslant 0$) rezultă că nu mai există activităţi eficiente şi prin urmare soluţia de bază analizată este optimă.

În cazul problemelor de minim indicatorii z_j se pot interpreta ca venituri, exprimate în preţurile umbră ale restricţiilor, ce se obţin prin realitarea unei unităţi din produsul j. Prin urmare dacă există diferenţe pozirive ($\Delta z_j > 0$), activităţile a_j corespunzătoare sînt eficiente deoarece se realizează un venit unitar mai mare decît costul unitar de fabricaţie ($z_j > c_j$). De aceea va intra în bază activitatea a_k corespunzătoare diferenţei maxime iar soluţia de bază se consideră optimă atunci cînd toate diferenţele Δz_j sînt nepozitive ($\Delta z_j \leqslant 0$).

2.9. Alte metode de rezolvare a problemelor de programare liniară

2.9.1. Algoritmul simplex dual

Fie **B** o bază pentru problema primală de minim astfel încît sînt verificate condiţiile $z_j - c_j \leqslant 0$ (sau $z_j - c_j \geqslant 0$ dacă este o problemă de maximum).

Scanned with OKEN Scanner

Vectorul $\mathbf{C}'_{\mathbf{B}}\mathbf{B}^{-1}$ constituie un program dual pentru problema primală și în cazul unei probleme de minimizat, vom avea :

$$\mathbf{C}'_{\mathbf{B}}\mathbf{B}^{-1}a_j = z_j \leqslant c_j \; (j = 1,2,\ldots,n)$$

Vectorul $\mathbf{X}^{\mathbf{B}}$ corespunzător îl numim așa după cum am văzut program (soluție) dual realizabil de bază.

Dacă în plus $\mathbf{X}^{\mathbf{B}} \geqslant 0$, adică este un program primal realizabil, atunci baza $\mathbf{B}$ este optimă pentru problema primală și deci $\mathbf{C}'_{\mathbf{B}}\mathbf{B}^{-1}$ constituie un program optimal al dualei.

Algoritmul simplex dual pornește de la o soluție dual realizabilă dar care nu este realizabilă (pentru problema primală) și urmărește eliminarea din bază a variabilelor negative menținînd, în fiecare iterație, verificat criteriul de optimalitate.

Să presupunem că pornim cu problema primală

$$\min \; [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$

$$\mathbf{AX} = \mathbf{b}$$

$$\mathbf{X} \geqslant 0$$

a cărei dală este

$$\max \; [\mathbf{G}(u)] = \mathbf{b}'\mathbf{U}$$

$$\mathbf{U}'\mathbf{A} \leqslant \mathbf{C}'$$

$$\mathbf{U} \text{ oarecare}$$

Se constată imediat că pentru orice cuplu $(\mathbf{X},\mathbf{U})$ de programe primal și dual avem

$$\overline{\mathbf{U}}'\mathbf{b} \leqslant \mathbf{C}'\overline{\mathbf{X}}$$

Să presupunem că se cunoaște o bază $\mathbf{B}$ pentru problema primală :

$$\mathbf{B} = (a_{j_1}, a_{j_2}, \ldots, a_s, \ldots, a_{j_m}), \; \{j_1, j_2, \ldots, s, \ldots, j_m\} = \overline{\mathbf{J}}_{\mathbf{B}}$$

astfel încît $\mathbf{U}' = \mathbf{C}'_{\mathbf{B}}\mathbf{B}^{-1}$ constituie un program dual, adică

$$\overline{\mathbf{U}}'a_j \leqslant c_j \; (\mathrm{j} \in \mathbf{J}_{\mathbf{B}})$$

sau altfel scris

$$z_j - c_j \leqslant 0 \quad (j \in \mathbf{J}_{\mathbf{S}})$$

Presupunem că soluția primală de bază $\overline{X}^{\mathbf{B}} = \mathbf{B}^{-1}\,\mathbf{b}$ este dual realizabilă.

Dacă $\mathbf{U}$ nu este soluție optimală atunci $\overline{X}^{\mathbf{B}}$ nu este un program pentru problema primală, adică există cel puțin un indice s astfel încît $\bar{x}_s \leqslant 0$.

Reciproc dacă există $\bar{x}_l < 0$ atunci putem găsi o altă soluție $\widetilde{\mathbf{U}}$ mai bună decît $\overline{\mathbf{U}}$.

Fie β_l linia de indice l a matricii $\mathbf{B}^{-1}$ și să notăm $\widetilde{\mathbf{U}}' = \overline{\mathbf{U}}' - \theta\beta_l$. Înmulțind la dreapta cu $\mathbf{b}$ obținem :

$$\widetilde{\mathbf{U}}'\mathbf{b} = \overline{\mathbf{U}}'\mathbf{b} - \theta\,\beta_l\,\mathbf{b} = \overline{\mathbf{U}}'\mathbf{b} - \theta\,\bar{x}_l$$

Scanned with OKEN Scanner

De aici rezultă că pentru orice $\theta > 0$ obținem $\tilde{U}'b > \bar{U}'b$ (adică o valoare mai bună a funcției obiectiv).

Pe de altă parte

$$\tilde{U}'a_j = \bar{U}'a_j - \theta\beta_l a_j \quad (j = 1,2,\ldots,n)$$

adică :

$$\tilde{U}'a_j = \bar{U}'a_j - \theta y_{lj} = z_j - \theta y_{lj} \quad j \in J_S$$

$$\tilde{U}'a_s = \bar{U}'a_s = z_s = c_s \quad s \in J_S - \{l\}$$

$$\tilde{U}'a_l = \bar{U}'a_l - \theta = z_l - \theta = c_l - \theta$$

Sînt posibile acum două cazuri :

a) $y_{lj} \geqslant 0 \quad j \in J_S$

În acest caz $\tilde{U}'$ este un program dual oricare ar fi $\theta > 0$ și dacă facem pe $\theta \to \infty$ în relația $\tilde{U}'b = \bar{U}'b - \theta x_l$ rezultă că duala nu admite un maximum.

Așadar în acest caz problema primală nu admite soluție optimă.

b) $y_{lj} < 0$ pentru cel puțin un indice $j \in J_S$.

În acest caz $\tilde{U}'$ rămîne program dual dacă și numai dacă $z_j - \theta y_{lj} \leqslant c_j$, $(j \in J_S)$ deci pentru

$$\theta \leqslant \min_{j/y_{lj}<0} \left[\frac{z_j - c_j}{y_{lj}}\right], \quad j \in J_S$$

Dacă se ia

$$\theta = \frac{z_k - c_k}{y_{lk}} = \min_{j/y_{lj}<0} \left[\frac{z_j - c_j}{y_{lj}}\right], j \in J_S$$

atunci :

$$\tilde{U}'a_k = z_k - \theta y_{lk} = z_k - \frac{z_k - c_k}{y_{lk}} y_{lk} = c_k$$

Să considerăm matricea $\tilde{B}$ dedusă din B prin înlocuirea vectorului a_l cu vectorul a_k.

Deoarece $y_{lk} \neq 0$ rezultă că $\tilde{B}$ este o bază cu mulțimea indicilor vectorilor coloană $\{J_{\tilde{B}} = J_B - \{l\} + \{k\}$ atunci $\tilde{U}'\tilde{B} = C'_{\tilde{B}}$ și deci $\tilde{U}' = C'_{\tilde{B}}\tilde{B}^{-1}$. Să notăm $\tilde{X}^{\tilde{B}} = \tilde{B}^{-1}b$ noua soluție de bază.

Dacă $\tilde{X}^{\tilde{B}} \geqslant 0$ această soluție de bază va constitui un program optimal de bază a problemei duale.

Dacă $\tilde{X}^{\tilde{B}}_s < 0$ pentru cel puțin un indice $s \in \tilde{J}_{\tilde{B}}$ se continuă procedeul iterativ obținîndu-se o nouă soluție duală $\tilde{\tilde{U}}$ mai bună decît $\tilde{U}$.

La fiecare iterație dacă se elimină (provizoriu) cazul cînd $\theta = 0$, valoarea funcției obiectiv a dualei $[G(u)]$ este strict crescătoare și deci nici o bază nu se poate repeta.

Pe de altă parte pentru a putea începe aplicarea algoritmului trebuie să cunoaștem un program dual, ceea ce exclude posibilitatea ca problema primală să nu aibă minim finit. Cum numărul bazelor este finit, după un număr finit de iterații se ajunge în mod necesar (cel mai tîrziu cînd s-a trecut prin toate bazele) la una din situațiile :

$\bar{X}^B \geqslant 0$ și deci programul este optimal

sau

$\bar{x}_l^B < 0$ $(l \in J_B)$ și $y_{lj} \geqslant 0$ $(j \in J_S)$ și problema primală nu are soluție.

Pentru a obține la fiecare iterație creșterea maximă a funcției obiectiv a formei duale trebuie determinat indicele l astfel încît :

$$\bar{x}_l \left(\frac{z_k - c_k}{y_{lk}} \right)$$

să fie maximă.

În practică însă, așa cum s-a procedat și la algoritmul simplex primal se înlocuiește această condiție printr-una mai simplă :

$$\bar{x}_l = \min_{s/\bar{x}_s < 0} \{\bar{x}_s\} \quad s \in J_B$$

În fiecare iterație crește valoarea funcției obiectiv a problemei duale și deci și a problemei primale.

Pașii algoritmului simplex dual sînt :

a) Se determină o bază $\mathbf{B}$ a problemei primale astfel încît $z_j - c_j \leqslant 0$ (sau $z_j - c_j \geqslant 0$ în cazul problemei de maximum)

b) Se testează soluția optimă $\bar{X}^B = B^{-1} b$.

Dacă $\bar{X}^B \geqslant 0$ programul este optim.

În caz contrar fie $I_l^B = \{s \in \bar{J}_B : \bar{x}_s < 0\}$.

c) Se testează y_{sj} $(s \in I_l^B,\ j \in J_S)$.

Dacă $y_{sj} \geqslant 0$ pentru cel puțin un $s \in I_l^B$ și orice $j \in J_S$ problema primală nu are programe.

Dacă există $y_{sj} < 0$ pentru cel puțin un $j \in J_S$ și orice $s \in I_l$ se determină l din relația :

$$\bar{x}_l = \min_{s \in I_l} \{x_s\}$$

care constituie criteriul de ieșire din bază și apoi se determină indicele k din relația

$$\frac{z_k - c_k}{y_{lk}} = \min_{j/y_{lj}<0} \left[\frac{z_j - c_j}{y_{lj}} \right]$$

$$j \in J_B$$

care constituie criteriul de intrare în bază. În cazul problemelor de maximum se alege raportul cu valoare maximă.

Scanned with OKEN Scanner

d) Se formează noua bază $\widetilde{\mathbf{B}}$ dedusă din $\mathbf{B}$ prin înlocuirea vectorului a_l cu vectorul a_k şi se calculează noul tabel simplex după regula cunoscută la algoritmul simplex primal.

Algoritmul se repetă de la pasul *b* pînă cînd se obţine soluţia optimă.

În cazul problemelor de maxim se înlocuieşte operaţia de minim cu cea de maxim în relaţia care defineşte criteriul de intrare în bază.

Acest algoritm va fi utilizat în aplicaţii la „Programarea parametrică".

2.9.2. Metoda simplex revizuită (modificată)

Pentru rezolvarea pe calculator a problemelor de programare liniară s-au elaborat variante ale algoritmului simplex, prin care se urmăreşte pe de o parte reducerea volumului de informaţii stocate în memoria calculatorului, iar pe de altă parte reducerea numărului de calcule şi deci creşterea vitezei de convergenţă a algoritmului.

Metoda simplex revizuită [*16*] care constituie o perfecţionare a algoritmului simplex obişnuit în sensul criteriilor de mai sus, se fundamentează pe ideea trecerii de la o soluţie posibilă la alta folosind matricea inversă $\mathbf{B}_h^{-1}$ a bazei $\mathbf{B}_h$ corespunzătoare iteraţiei h şi informaţiile iniţiale, matricele $\mathbf{A}$, $\mathbf{b}$ şi $\mathbf{C}$, date în problema de programare liniară. Relaţiile de bază utilizate în acest scop sînt (vezi paragraful 2.5) :

$$y_j^{\mathbf{B}} = \mathbf{B}^{-1} \cdot a_j \quad (j = 1,\ 2, \ldots, n)$$

$$x^{\mathbf{B}} = \mathbf{B}^{-1} \cdot \mathbf{b}$$

$$z_j^{\mathbf{B}} = \mathbf{C}_{\mathbf{B}}' \cdot \mathbf{B}^{-1} \cdot a_j$$

(pentru simplificarea scrierii s-a renunţat la indicele h).

Rezultă că toate elementele tabelului simplex corespunzător unei iteraţii h se pot deduce prin operaţii care utilizează informaţiile iniţiale şi matricea $\mathbf{B}^{-1}$.

În cadrul metodei simplex revizuite trecerea de la o bază la alta presupune transformarea, prin utilizarea formulelor eliminării complete, numai a elementelor matricei inverse.

Pentru prezentarea algoritmului simplex modificat vom considera următoarea problemă de programare liniară scrisă sub formă standard :

Să se determine minimul funcţiei

$$\mathbf{Z}(x) = \mathbf{C}'\mathbf{X}$$

(în condiţiile

$$\mathbf{AX} + \mathbf{I}\,\mathbf{X}_a = \mathbf{b} \qquad (2.21)$$

$$\mathbf{X} \geqslant 0$$

unde $\mathbf{X}_a$ este vectorul variabilelor de ecart sau artificiale $x_{n+1}, x_{n+2}, \ldots, x_{n+m}$, care formează soluţia de bază iniţială.

Obţinerea soluţiei optime presupune determinarea indicatorilor $\Delta z_j = z_j - c_j$ care caracterizează eficienţa activităţilor a_j, $(j = 1,\ 2, \ldots, n)$ în fiecare iteraţie. Dar, după cum s-a arătat mai înainte (vezi paragrafele

Scanned with OKEN Scanner

2.5 și 2.8.), indicatorii z_j se pot calcula prin relațiile :

$$z_j = \sum_{i=1}^{m} c_i^{\mathbf{B}} y_{ij} \quad (j = 1,\ 2,\ldots,n)$$

sau

$$z_j = \sum_{i=1}^{m} u_i a_{ij}$$

unde u_i reprezintă prețurile umbră ale resurselor sau multiplicatorii simplex.

Deoarece aplicarea primei relații necesită cunoașterea tuturor elementelor y_{ij} ale tabelului simplex, la algoritmul simplex revizuit se folosește cea de a doua relație. Reamintim că vectorul $\mathbf{U} = (u_1,\ u_2,\ldots,u_m)$ se definește prin $\mathbf{U} = \mathbf{C}'_{\mathbf{B}}\mathbf{B}^{-1}$ și prin urmare vectorul $\mathbf{Z} = (z_1, z_2,\ldots,z_n)$ rezultă din $\mathbf{Z} = \mathbf{C}'_{\mathbf{B}}\mathbf{B}^{-1}\mathbf{A}$.

Indicatorii Δz_j se pot determina din expresia :

$$(\Delta z_1,\ \Delta z_2,\ldots,\ \Delta z_n) = (u_1,\ u_2,\ldots,\ u_m,\ 1) \begin{pmatrix} a_{11} & a_{12}\ldots a_{1n} \\ a_{21} & a_{22}\ldots a_{2n} \\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot \\ a_{m1} & a_{m2}\ldots a_{mn} \\ -c_1 & -c_2\ldots -c_n \end{pmatrix}$$

De aceea modelului matematic al problemei de programare liniară i se atașează restricția $m + 1$:

$$-c_1x_1 - c_2x_2 - \ldots - c_nx_n + \mathbf{Z} = 0 \qquad (2.22)$$

care permite calculul diferențelor Δz_j în cadrul algoritmului simplex revizuit.

În vederea eliminării variabilelor artificiale, problemei inițiale i se adaugă încă o restricție care, dacă considerăm ipoteza că toate variabilele x_{n+i} $(i = 1,\ 2,\ldots,m)$ sînt artificiale, se construiește astfel :

$$a_{m+2,1}\,x_1 + a_{m+2,2}\,x_2 + \ldots + a_{m+2,n}\,x_n + x_{n+m+2} = b_{m+2} \qquad (2.23)$$

unde

$$a_{m+2,j} = -\sum_{i=1}^{m} a_{ij} \qquad j = 1,2,\ldots,n$$

$$b_{m+2} = -\sum_{i=1}^{m} b_i$$

Variabila* x_{n+m+2} reprezintă suma luată cu semnul minus, a variabilelor artificiale, adică :

$$x_{n+m+2} = -(x_{n+1} + x_{n+2} + \ldots + x_{n+m}) \qquad (2.24)$$

* Se consideră că $x_{n+m+1} = \mathbf{Z}$.

Scanned with OKEN Scanner

Trebuie precizat că dacă problema (2.21) conține variabile de ecart, sau vectori a_j unitari cu $c_j = 0$ $(j = 1, 2, \ldots, n)$, aceștia pot fi utilizați în baza inițială a algoritmului simplex modificat cu condiția ca ecuațiile în care apar să fie excluse de la calculul coeficienților restricției $m + 2$, adică :

$$a_{m+2,j} = -\sum_{i \notin J_B} a_{ij}\,;\quad b_{m+2} = -\sum_{i \notin J_B} b_i$$

Prin adăugarea restricțiilor (2.22) și (2.23) la problema originară se obține o problemă extinsă de programare liniară a cărei formă este :

$$\min \mathbf{Z}$$

în condițiile :

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + x_{n+1} = b_1\\
&a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + x_{n+2} = b_2\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + x_{n+m} = b_m\\
&-c_1x_1 - c_2x_2 - \ldots - c_nx_n + \mathbf{Z} = 0\\
&a_{m+2,1}x_1 + a_{m+2;2}x_2 + \ldots + a_{m+2,n}x_n + x_{n+m+2} = b_{m+2}\\
&x_j \geqslant 0\ (j = 1,2,\ldots n)
\end{aligned}$$

Matricea corespunzătoare sistemului de restricții a problemei extins are următoarea structură :

$$\begin{pmatrix}
a_{11} & a_{12} & \ldots & a_{1n} & 1 & 0 & \ldots & 0 & 0 & 0\\
a_{21} & a_{22} & \ldots & a_{2n} & 0 & 1 & \ldots & 0 & 0 & 0\\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots\\
a_{m1} & a_{m2} & \ldots & a_{mn} & 0 & 0 & \ldots & 1 & 0 & 0\\
-c_1 & -c_2 & \ldots & -c_n & 0 & 0 & \ldots & 0 & 1 & 0\\
a_{m+2,1} & a_{m+2,2} & \ldots & a_{m+2,n} & 0 & 0 & \ldots & 0 & 0 & 1
\end{pmatrix} = (\bar{\mathbf{A}} : \mathbf{I})$$

Primele n coloane formează matricea $\bar{\mathbf{A}}$ iar ultimele $m + 2$ constituie o matrice unitate de ordinul $m + 2$.

Soluția posibilă de bază a problemei extinse este formată din variabile artificiale x_{n+i} $(i = 1,2,\ldots,n)$ precum și din variabilele $\mathbf{Z}$ și x_{n+m+2}.

După cum s-a arătat la începutul acestui paragraf, în cadrul algoritmului simplex revizuit se transformă pe baza formulelor eliminării complete numai elementele matricei $\mathbf{B}^{-1}$, de aceea tabelele simplex folosite la rezolvarea problemei de programare liniară au structura dată în tabelul 2.17.

În tabelul simplex inițial $(h = 0)$, vectorii bazei sînt $a_{n+1}, a_{n+2}\ldots, a_{n+m}$, $\mathbf{Z}$ și a_{n+m+2} ; soluția de bază are componentele* $x_{n+i} = b_i$ $(i = 1, 2, \ldots, m + 2)$; iar matricea $\mathbf{B}_0^{-1} = \mathbf{I}$.

* Facem precizarea că $b_{m+1} = 0$.

Scanned with OKEN Scanner

Tabelul 2. 17.

Nr. rînd	Vectorii bazei a_i	Valoarea variabilelor din baza $\bar{x}_i$	Elementele matricei B_h^{-1}	Componentele vectorului care intră în bază y_{ik}
1		$\bar{x}_1$	β_{11} β_{12} . . . β_{1l} . . . β_{1m} 0 0	y_{1k}
2		$\bar{x}_2$	β_{21} β_{22} . . . β_{2l} . . . β_{2m} 0 0	y_{2k}
.		.	. . . . .	.
.		.	. . . . .	.
.		.	. . . . .	.
l		$\bar{x}_l$	β_{l1} β_{l2} . . . β_{ll} . . . β_{lm} 0 0	y_{lk}
.		.	. . . . .	.
.		.	. . . . .	.
.		.	. . . . .	.
m		$\bar{x}_m$	β_{m1} β_{m2} . . . β_{ml} . . . β_{mn} 0 0	y_{mk}
m + 1	Z	$\bar{x}_{n+m+1}$	$\beta_{m+1,1}$ $\beta_{m+1,2}$ . . . $\beta_{m+1,l}$ . . . $\beta_{m+1,m}$ 1 0	$y_{m+1,k}$
m + 2	x_{n+m+2}	$\bar{x}_{n+m+2}$	$\beta_{m+2,1}$ $\beta_{m+2,2}$ . . . $\beta_{m+2,l}$ . . . $\beta_{m+2,m}$ 0 1	$y_{m+2,k}$

Primele m linii şi m coloane ale matricii B_h^{-1} reprezintă inversa bazei formată din m vectori aleşi dintre vectorii a_i ($i = 1, 2, \ldots, n$) pe baza unor criterii ce se vor prezenta ulterior, iar ultimele două linii servesc la determinarea vectorului ce urmează a fi introdus în bază. Coloanele $m + 1$ şi $m + 2$ ale matricii B_h^{-1} nu se modifică în cadrul transformărilor prevăzute de algoritmul simplex revizuit deoarece variabilele $\mathbf{Z}$ şi x_{m+n+2} nu se elimină din bază. Pe de altă parte, cei doi vectori unitari ce formează aceste coloane permit calculul diferenţelor $\Delta z_j = z_j - c_j$.

Pentru obţinerea soluţiei optime cu ajutorul algoritmului simplex revizuit se parcurg două etape. În prima etapă se urmăreşte eliminarea din bază a variabilelor artificiale X_{n+i} ($i=1, 2 \ldots, m$). Avînd în vedere relaţia (2.24), rezultă că soluţia iniţială pentru problema (2.21), indiferent că este o problemă de maxim sau de minim, se determină prin maximizarea* variabilei x_{n+m+2}.

Atunci cînd $x_{n+m+2} = 0$, toate variabilele artificiale au fost eliminate din bază, obţinîndu-se soluţia iniţială a problemei (2.21). În cazul în care $\max (x_{n+m+2}) < 0$, soluţia de bază conţine cel puţin o variabilă artificială iar problema (2.21) nu are soluţie.

Dacă în prima etapă se obţine $x_{n+m+2} = 0$ se trece la etapa a doua care constă în minimizarea variabilei $\mathbf{Z}$. Se observă imediat din relaţia (2.22) că $\mathbf{Z} = \mathbf{C'X}$ şi prin urmare minimizarea variabilei $\mathbf{Z}$ este echivalentă cu minimizarea funcţiei scop a problemei originare.

La problemele în care se cere maximizarea funcţiei obiectiv a problemei originare, pe baza aceleiaşi explicaţii de mai sus, se va maximiza variabila $\mathbf{Z}$ a problemei extinse.

În continuare se prezintă calculele ce se efectuează în cadrul fiecărei etape.

---

* Se observă că dacă în relaţia (2.23) se introduce variabile x_{n+m+2} cu semnul minus, atunci $x = \sum_{i=1}^{m} x_{n+i}$ realizîndu-se analogie cu prima etapă a metodei celor două faze, adică minimizarea sumei variabilelor artificiale.

Scanned with OKEN Scanner

Etapa I

I.1. Prima operație constă în calcularea componentelor $\Delta^I z_j = z_j - c_j$ ale vectorului $\Delta \mathbf{Z}$ pe baza relației :

$$\Delta^I \mathbf{Z} = \beta_{(m+2)} \bar{\mathbf{A}} \tag{2.25}$$

unde $\beta_{(m+2)}$ este vectorul format din elementele liniei $m+2$ a matricei $\mathbf{B}_h^{-1}$ ($h = 0,1,2,\ldots$). Nu este greu de văzut că relația (2.25) răspunde scopului propus întrucît primele m elemente ale vectorului $\beta_{(m+2)}$ sînt tocmai prețurile umbră u_i ($i = 1, 2, \ldots, m$) atașate restricțiilor, iar datorită faptului că în etapa I $c_j = 0$ ($j = 1, 2, \ldots, n$), componentele vectorului $\Delta^I \mathbf{Z}$ reprezintă diferențele $\Delta^I \mathbf{Z}_j = z_j - c_j$. După cum s-a arătat, aceste diferențe caracterizează eficiența activităților a_j ($j = 1, 2, \ldots, n$) în funcție de vectorii care formează baza în fiecare iterație h.

I.2. În cazul cînd $\Delta^I \mathbf{Z}_j \geqslant 0$ ($j = 1, 2, \ldots, n$), nici un vector artificial nu se elimină din bază iar $\max (x_{n+m+2}) < 0$ și de aceea problema (2.21) nu are soluție.

Dacă cel puțin unul dintre indicatorii $\Delta z_j < 0$, atunci va intra în noua bază vectorul a_k care se stabilește prin relația :

$$\Delta^I \mathbf{Z}_k = \min \Delta^I z_j$$

I.3. Se calculează componentele vectorului a_k ce urmează a fi introdus în bază :

$$y_k = \mathbf{B}^{-1} \cdot \bar{a}_k \tag{2.26}$$

unde $\bar{a}_k$ este coloana k din matricea $\bar{\mathbf{A}}$. Cunoscînd componentele y_{ik} putem determina vectorul a_l ce iese din bază utilizînd relația :

$$\theta_l = \min_i \left(\frac{x_i}{y_{ik}} \right) (i = 1, 2, \ldots, m) \quad (y_{ik} > 0) \tag{2.27}$$

I.4. Se aplică formulele (2.6) și (2.7) pentru a calcula valoarea $\bar{x}_i$ a variabilelor din noua bază precum și componentele matricei $\mathbf{B}^{-1}$

Operațiile din cadrul primei etape se repetă pînă cînd se constată că problema (2.21) nu are soluție sau pînă în momentul cînd $x_{n+m+2} = 0$.

Etapa II-a

II.1. Se calculează diferențele $\Delta^{II} Z_j = z_j - c_j$ folosind relația :

$$\Delta^{II} \mathbf{Z} = \beta_{(m+1)} \cdot \bar{\mathbf{A}}$$

unde $\beta_{(m+1)}$ este vectorul format din elementele liniei $m+1$ a matricei $\mathbf{B}^{-1}$.

Primele m elemente ale acestei linii sînt prețurile umbră u_i ($i = 1, 2, \ldots, m$) ale resurselor. În cadrul etapei a doua la calculul diferențelor $\Delta^{II} \mathbf{Z}_j$ intervin și coeficienții c_j ($j = 1, 2 \ldots, n$) ai funcției obiectiv deoarece $b_{m+1,m+1} = 1$.

II.2. Dacă $\Delta^{II} z_j \leqslant 0$ atunci s-a obținut valoarea minimă a variabilei $\mathbf{Z}$.

Scanned with OKEN Scanner

În cazul în care există $\Delta^{II}z_j > 0$ se va introduce în bază vectorul a_k căruia îi corespunde diferența $\Delta^{II}z_j$ maximă, adică :

$$\Delta^{II}Z_k = \max\ \Delta^{II}Z_j \quad (j = 1,\ 2,\ldots,n)$$

II.3. Se calculează componentele vectorului a_k și se stabilește vectorul ce iese din bază folosind relațiile (2.26) și (2.27).

II.4. Se construiește tabelul simplex corespunzător bazei noi aplicînd formulele (2.6) și (2.7).

Soluția optimă a problemei (2.21) se obține cînd $\Delta^{II}z_j \leqslant 0$.

Pentru problemele de maxim, criteriul de optimalitate este $\Delta^{II}z_j \geqslant 0$.

Analizînd algoritmul simplex revizuit se constată că la fiecare iterație trebuie să se rețină în memoria calculatorului, în afară de informațiile inițiale, numai matricea $\mathbf{B}^{-1}$ și vectorii x și y_k. În acest fel se evită calcularea și memorarea întregului tabel simplex. Pe de altă parte, datorită faptului că la algoritmul simplex revizuit se folosesc operații care implică doar elementele nenule ale matricei **A**, prin aranjarea compactă a acestora se obține o reducere substanțială a memoriei calculatorului.

Metoda simplex modificată, combinată cu metoda de calcul a matricei $\mathbf{B}^{-1}$ sub formă de produs de matrice, se folosește la elaborarea unor programe eficiente de rezolvare a problemelor de programare liniară cu ajutorul calculatoarelor electronice.

Exemplul 2.13. Vom rezolva problema primală dată în exemplul 2.12 folosind algoritmul simplex revizuit. Problema extinsă corespunzătoare acestui model are forma :

$$\max \mathbf{Z}$$

în condițiile

$$\begin{aligned}
x_1 + 3x_2 + 4x_3 + x_4 &= 15\\
2x_1 + 5x_2 + x_3 + x_5 &= 10\\
4x_1 + x_2 + 2x_3 + x_6 &= 25\\
x_2 - x_7 + x_8 &= 1\\
-3x_1 - 2x_2 - 6x_3 + \mathbf{Z} &= 0\\
-x_1 - 4x_2 - 4x_3 + x_7 + x_{10} &= -16
\end{aligned}$$

$$x_j \geqslant 0 \ (j = 1, 2,\ldots,8).$$

Matricea $\bar{\mathbf{A}}$ corespunzătoare modelului este :

$$\bar{\mathbf{A}} = \begin{pmatrix}
1 & 3 & 4 & 0\\
2 & 5 & 1 & 0\\
4 & 1 & 2 & 0\\
0 & 1 & 0 & -1\\
-3 & -2 & -6 & 0\\
-1 & -4 & -4 & 1
\end{pmatrix}$$

Scanned with OKEN Scanner

În tabelele 2.18 sînt înregistrate datele rezultate în urma aplicării algoritmului simplex revizuit. Tabelul T_0 conține datele problemei extinse. Calculele efectuate în cadrul celor trei iterații necesare obținerii soluției optime sînt prezentate mai jos.

Iterația 1

$$\Delta \mathbf{Z}^I = (0,0,0,0,0,1)\cdot \bar{\mathbf{A}} = (-1,-4,-4,1)$$

$$\min \Delta \mathbf{Z}^I = \Delta \mathbf{Z}_3^I$$

Ca urmare va intra în bază vectorul a_3 și de aceea se calculează $y_3 = \mathbf{B}^{-1}\bar{a}_3$

$$y_3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 4 \\ 1 \\ 2 \\ 0 \\ -6 \\ -4 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ 2 \\ 0 \\ -6 \\ -4 \end{pmatrix}$$

Componentele vectorului y_3 se trec în ultima coloană a tabelului T_0 și pe baza relației (2.27) se stabilește că vectorul a_4 iese din bază.

Prin aplicarea formulelor (2.6) și (2.7) se calculează componentele $\bar{x}_i$ ale noii soluții de bază și componentele matricei $\mathbf{B}$, obținîndu-se rezultatele din tabelul T_1.

Iterația a 2-a

$$\Delta \mathbf{Z}^I = (1,0,0,0,0,1).\bar{\mathbf{A}} = (0,-1,0,1)$$

$\min \Delta \mathbf{Z}^I = \Delta \mathbf{Z}_2^I$ (deci intră în baza vectorului a_2)

$$y_2 = \begin{pmatrix} 1/4 & 0 & 0 & 0 & 0 & 0 \\ -1/4 & 1 & 0 & 0 & 0 & 0 \\ -2/4 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 6/4 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 5 \\ 1 \\ 1 \\ -2 \\ -4 \end{pmatrix} = \begin{pmatrix} 3/4 \\ 17/4 \\ -2/4 \\ 1 \\ 10/4 \\ -1 \end{pmatrix}$$

Componentele vectorului y_2 se trec în ultima coloană a tabelului T_1, constatîndu-se că iese din bază vectorul a_8. Noua soluție de bază precum și matricea $\mathbf{B}^{-1}$ sînt trecute în tabelul T_2. Se constată că variabila $x_{10} = 0$ și de aceea se trece la etapa a II-a.

Iterația a 3-a

$$\Delta \mathbf{Z}^{II} = (6/4,0,0,-10/4,1,0)\cdot \bar{\mathbf{A}} = (-6/4,0,0,10/4)$$

Deoarece în problema inițială se cere maximizarea funcției $\mathbf{Z}(x)$, atunci vectorul ce intră în bază se determină pe baza relației :

Scanned with OKEN Scanner

$$\min \Delta Z^{II} = \Delta Z_7^{II}$$

$$y_1 = \begin{pmatrix} 1/4 & 0 & 0 & -3/4 & 0 & 0 \\ -1/4 & 1 & 0 & -17/4 & 0 & 0 \\ -2/4 & 0 & 1 & 2/4 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 6/4 & 0 & 0 & -10/4 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 4 \\ 0 \\ -3 \\ -1 \end{pmatrix} = \begin{pmatrix} 1/4 \\ 7/4 \\ 14/4 \\ 0 \\ -6/4 \\ 0 \end{pmatrix}$$

După introducerea în bază a vectorului a_1 în locul vectorului a_5 se obţine soluţia dată în tabelul T_3. Deoarece

$$\Delta Z^{II} = (9/7,\ 6/7,\ 0,\ -43/7,\ 1,0)\,\bar{A} = (0, 0, 0, 43/7) \geqslant 0$$

s-a obţinut soluţia optimă căutată.

Tabelul 2.18.

		B	$\bar{x}_i$	a_4	a_5	a_6	a_8	Z	a_{10}	y_{ik}
T_0	1	a_4	15	1	0	0	0	0	0	①
	2	a_5	10	0	1	0	0	0	0	1
	3	a_6	25	0	0	1	0	0	0	2
	4	a_8	1	0	0	0	1	0	0	0
	5	z	0	0	0	0	0	1	0	−6
	6	a_{10}	−16	0	0	0	0	0	1	−4
T_1	1	a_3	15/4	1/4	0	0	0	0	0	3/4
	2	a_5	25/4	−1/4	1	0	0	0	0	17/4
	3	a_6	70/4	−2/4	0	1	0	0	0	−2/4
	4	a_8	1	0	0	0	1	0	0	①
	5	z	45/2	6/4	0	0	0	1	0	10/4
	6	a_{10}	−1	1	0	0	0	0	1	−1
T_2	1	a_3	3	1/4	0	0	−3/4	0	0	1/4
	2	a_5	2	−1/4	1	0	−17/4	0	0	⑦/4
	3	a_6	18	−2/4	0	1	2/4	0	0	14/4
	4	a_2	1	0	0	0	1	0	0	0
	5	z	20	6/4	0	0	−10/4	1	0	−6/4
	6	a_{10}	0	1	0	0	1	0	1	0
T_3	1	a_3	19/7	2/7	−1/7	0	−1/7	0	0	
	2	a_1	8/7	−1/7	4/7	0	−17/7	0	0	
	3	a_6	14	0	−2	1	9	0	0	
	4	a_2	1	0	0	0	1	0	0	
	5	z	152/7	+9/7	6/7	0	−43/7	1	0	
	6	a_{10}	0	1	0	0	1	0	1	

2.10. Reoptimizarea şi parametrizarea în programarea liniară

După cum s-a arătat mai înainte, problemele de programare liniară conţin, în afară de variabilele x_j al căror nivel trebuie determinat, o serie de mărimi constante concretizate în componentele vectorilor **b**, **C**,

Scanned with OKEN Scanner

şi ai matricei **A**. Este evident că valoarea practică a soluţiei optime depinde de precizia cu care s-a determinat nivelul acestor constante. Modelele de programare liniară analizate mai înainte sînt modele statice, ele nu surprind aspectele dinamice ale fenomenelor economice. Dar în activitatea practică de conducere planificată a activităţilor economice apar schimbări frecvente ale condiţiilor care stau la baza modelului de programare liniară. Ca urmare a apărut necesitatea elaborării unor metode matematice care să permită abordarea dinamică a fenomenelor economice.

Reoptimizarea şi parametrizarea problemelor de programare liniară tratează doar aspectele legate de variaţia elementelor, „constante" ale modelelor ataşate fenomenelor economice.

Reoptimizarea sau postoptimizarea constă în recalcularea soluţiei optime a unei probleme de programare liniară în cazul cînd se modifică unele dintre condiţiile iniţiale.

Prin parametrizarea unei probleme de programare liniară se urmăreşte determinarea variantelor posibile de soluţii optime în cazul cînd unele dintre „constantele" modelului variază liniar sau neliniar în funcţie de unul sau mai mulţi parametri daţi.

Aceste aspecte vor fi tratate în detaliu în paragrafele următoare.

2.10.1. Reoptimizarea

Este ştiut faptul că pe parcursul executării planului de producţie pot apărea, faţă de programul optim determinat pe baza unor condiţii economice şi tehnice prestabilite, o serie de modificări ale unor elemente ce constituie suportul informaţional a modelelor de programare liniară. În continuare vom prezenta cîteva cazuri prin care intenţionăm să surprindem unele aspecte economice privind modificarea condiţiilor prestabilite.

Astfel, componentele vectorului termenilor liberi se pot modifica faţă de nivelul lor prestabilit ca urmare a schimbării capacităţii mijloacelor de muncă, a disponibilului de forţă de muncă sau materii prime şi materiale, a schimbării nivelului impus unor indicatori economici sau tehnici care caracterizează „calitatea" unui program optim.

Schimbarea tehnologiei de fabricaţie are efecte directe asupra costurilor de producţie şi asupra consumurilor specifice de resurse. Dacă aceste modificări nu au fost anticipate corect, soluţia modelului de programare liniară folosit în optimizarea programului de producţie nu corespunde condiţiilor reale în care se desfăşoară procesul economic.

Dacă în perioada de plan se asimilează un nou produs, în modelul de programare liniară trebuie introdusă o nouă variabilă şi ca urmare matricea **A** va conţine un non vector coloană. Înlocuirea unei surse de energie cu o alta a unui material cu un altul etc. presupune includerea în model a unor restricţii suplimentare.

În astfel de situaţii, se pune întrebarea dacă soluţia optimă obţinută pe baza condiţiilor iniţiale îşi mai păstrează caracterul de optimalitate şi după modificările acestora. Pentru a răspunde la întrebare, va trebui să se verifice soluţia în noile condiţii. Dacă prin verificare s-a ajuns la concluzia că soluţia nu mai este optimă, atunci se pune o nouă problemă, aceea de a găsi metode de reoptimizare a soluţiei, metode care să utilizeze rezul-

Scanned with OKEN Scanner

tatele anterioare, pentru ca în felul acesta să se micșoreze volumul calculelor cît mai mult posibil, chiar dacă acestea se fac pe un calculator electronic.

Aşadar, scopul reoptimizării este de a studia modificările care survin în soluţia optimă a unei probleme de programare liniară atunci cînd se modifică datele problemei. Importanţa unui astfel de studiu, rezidă în economia de muncă ce se poate face, utilizînd informaţia de care dispunem deja — adică soluţia optimă cunoscută.

Precizăm că în procesul de reoptimizare vom aborda separat următoarele aspecte :

— modificarea vectorului „termenilor liberi"
— modificarea vectorului „costurilor" $\mathbf{C}$;
— modificarea unui vector coloană al matricei $\mathbf{A}$;
— modificarea unui vector linie din $\mathbf{A}$;
— modificarea numărului restricţiilor sistemului ;
— modificarea numărului de variabile.

În vederea reoptimizării unei probleme de programare liniară, este necesar să se parcurgă următoarele etape :

1. Verificarea optimalităţii soluţiei în condiţiile schimbării datelor problemei ;

2. Determinarea soluţiei optime a problemei modificate (dacă soluţia problemei iniţiale nu mai este optimă).

Întrucît reoptimizarea se referă la soluţia optimă ($\overline{\mathbf{X}}^{\mathbf{B}}$) a problemei :

$$\min(\max)\ [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$

în condiţiile

$$\mathbf{A}\cdot\mathbf{X} = \mathbf{b}$$

$$\mathbf{X} \geqslant 0$$

în continuare vom trata toate cazurile de reoptimizare pe baza forme standard a problemei de programare liniară.

2.10.1.1. *Modificarea vectorului termenilor liberi*

Presupunînd că vectorul $\mathbf{b}$ se înlocuieşte cu $\tilde{\mathbf{b}} = \mathbf{b} + \delta\,\mathbf{b}$ atunci soluţia optimă $\overline{\mathbf{X}}^{\mathbf{B}}$ se modifică după cum urmează :

$$\tilde{\mathbf{X}}^{\mathbf{B}} = \mathbf{B}^{-1}\cdot\tilde{\mathbf{b}} = \mathbf{B}^{-1}(\mathbf{b} + \delta\mathbf{b}) = \overline{\mathbf{X}}^{\mathbf{B}} + \mathbf{B}^{-1}\cdot\delta\mathbf{b}$$

Întrucît diferenţele $\Delta\mathbf{Z} = \mathbf{C}'_{\mathbf{B}}\cdot\mathbf{B}^{-1}\cdot\mathbf{A} - \mathbf{C}'$ sînt independente de vectorul $\mathbf{b}$, noua soluţie $\tilde{\mathbf{X}}_{\mathbf{B}}$ este :

1) optimă dacă toate componentele soluţiei sînt nenegative ($\tilde{\mathbf{X}}^{\mathbf{B}} \geqslant 0$) ;

2) dual realizabilă dacă vectorul $\tilde{\mathbf{X}}^{\mathbf{B}}$ are şi componente negative.

În cel de al doilea caz pentru aflarea soluţiei optime, vectorul $\tilde{\mathbf{X}}^{\mathbf{B}}$ se introduce în coloana termenilor liberi din tabelul simplex final al problemei iniţiale şi se aplică algoritmul simplex dual.

Scanned with OKEN Scanner

Exemplul 2.14. Pentru elaborarea programului de dezvoltare a unei centrale industriale se pune problema stabilirii variantei optime de dotare cu mijloace de muncă achiziționate prin import. În tabelul 2.19 sînt sintetizate informații cu privire la capacitatea de producție, consumul de materii prime de tipul M*1* fabricate în țară și de materii prime de tipul M2 importate, precum și prețurile unitare de achiziție a trei tipuri de utilaje. Trebuie precizat că cele trei tipuri de utilaje se folosesc la fabricarea aceluiași produs.

Tabelul 2.19

Indicatorii	Tipuri de utilaje		
	1	2	3
Capacitatea anuală de producție a unui utilaj (unități fizice)	5	4	5
Consumul anual de materii prime de tipul *M1* pe un utilaj (unități fizice)	3	6	5
Consumul anual de materii prime de tipul M_2, pe un utilaj (unități fizice)	2	1	3
Prețul de achiziție a unui utilaj (unități monetare)	10	7	9

Analizînd evoluția cererii pentru produsul ce urmează a fi fabricat de aceste utilaje s-a ajuns la concluzia că volumul producției trebuie să atingă un nivel anual de cel puțin 700 unități fizice. Avînd în vedere că disponibilul de materii prime din țară și din import este de 500 unități fizice și respectiv 300 unități fizice, rezultă următorul model de programare liniară :

$$\min [\mathbf{Z}(x)] = 10x_1 + 7x_2 + 9x_3$$

în condițiile

$$5x_1 + 4x_2 + 5x_3 \geqslant 700$$

$$3x_1 + 6x_2 + 5x_3 \leqslant 500$$

$$2x_1 + x_2 + 3x_3 \leqslant 300$$

$$x_j \geqslant 0 (j = 1,2,3)$$

unde x_j reprezintă numărul de utilaje de tipul $j(j = 1,2,3)$ ce urmează a fi achiziționate.

Plecînd de la soluția inițială de bază $\overline{\mathbf{X}}' = (x_1, x_2, x_3, x_4, x_7\, x_5, x_6,) = (0, 0, 0, 0, 700, 500, 300)$ se obține soluția optimă din tabelul 2.20.

Dacă vectorul termenilor liberi devine $\tilde{\mathbf{b}}' = (700, 800, 500)$ atunci soluția corespunzătoare bazei $\mathbf{B} = (a_1, a_2, a_3)$ are structura :

$$\tilde{\mathbf{X}}^{\mathbf{B}} = \mathbf{B}^{-1} \cdot \tilde{\mathbf{b}} = \begin{pmatrix} 13/24 & -7/24 & -5/12 \\ 1/24 & 5/24 & -5/12 \\ -3/8 & 1/8 & 3/4 \end{pmatrix} \cdot \begin{pmatrix} 700 \\ 800 \\ 500 \end{pmatrix} = \begin{pmatrix} -125/2 \\ -25/2 \\ 425/2 \end{pmatrix}$$

Scanned with OKEN Scanner

Tabelul 2.20

B	c_B	$\bar{X}$	10	7	9	0	0	0	λ
			a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_1	10	325/3	1	0	0	−13/24	−7/24	−5/12	13/24
a_2	7	25/3	0	1	0	−1/24	5/24	−5/12	1/24
a_3	9	25	0	0	1	−3/8	1/8	3/4	−3/8
	z_j	1366,67	10	7	9	−7/3	−1/8	−1/3	7/3
	Δz_j	—	0	0	0	−7/3	−1/3	−1/3	7/3−λ

Avînd în vedere că vectorul $\tilde{X}^B$ are componente negative iar $\Delta Z_j \leqslant 0$, soluţia obţinută este dual realizabilă şi pentru optimizarea ei se poate aplica algoritmul simplex dual.

Componentele vectorului $\tilde{X}^B$ se introduc în coloana $\bar{X}$ a tabelului 2.20 (vezi T_1 din tabelele 2.21) şi se utilizează algoritmul prezentat în paragraful (2.9.1)

Tabelul 2.21

	B	c_B	$\bar{X}$	10	7	9	0	0	0	λ
				a_1	a_2	a_3	a_4	a_5	a_6	a_7
T_1	a_1	10	−125/2	1	0	0	−13/24	−7/24	−5/12	13/24
	a_2	7	−25/2	0	1	0	−1/24	5/24	−5/12	1/24
	a_3	9	425/2	0	0	1	3/8	1/8	3/4	−3/8
	z_j		1 200	10	7	9	−7/3	−1/3	−1/3	7/3
	Δz_j			0	0	0	−7/3	−1/3	−1/3	7/3−λ
T_2	a_6	0	150	−4/5	0	0	13/10	7/10	1	−13/10
	a_2	7	50	−1	1	0	1/2	1/2	0	−1/2
	a_3	9	100	9/5	0	1	−3/5	−2/5	0	3/5
	z_j		1 250	46/5	7	9	−19/10	−1/10	0	19/10
	Δz_j		—	−4/5	0	0	−19/10	−1/10	0	19/10−λ

Se va elimina din bază vectorul a_1 deoarece min $(-125/2, -25/2) = -125/2$. Pentru a determina vectorul ce urmează să intre în bază se utilizează relaţia :

$$\frac{z_k - c_k}{y_{lk}} = \min_{y_{lj}<0}\left(\frac{z_j - c_j}{y_{lj}}\right) \quad (j \in J_S)$$

Concretizînd această relaţie la exemplul analizat aici se obţine:

$$\min\left(\frac{7}{3}\cdot\frac{24}{13};\ \frac{1}{3}\frac{24}{7};\ \frac{1}{3}\frac{12}{5}\right) = \frac{4}{5}$$

Valoarea minimă astfel obţinută corespunde vectorului a_6, prin urmare a_6 va înlocui în bază vectorul a_1. Folosind relaţiile de transformare

Scanned with OKEN Scanner

(2.6) şi (2.7) se calculează elementele noului tabel simplex (vezi T_2 din tabelele 2.21). Întrucît toate diferenţele $\Delta z_j \leqslant 0$, noua soluţie este optimă.

Este interesant de remarcat că faţă de soluţia din tabelul 2.20 se realizează o reducere a cheltuielilor de achiziţionare a utilajelor egală cu 116,67 unităţi monetare.

Întrucît creşterea consumului de materii prime fabricate în ţară este de 300 unităţi fizice, iar a consumului de materii prime importate de 50 unităţi fizice*, eficienţa acestei soluţii se apreciază ţinînd seama de preţul de achiziţie a celor două materii prime.

2.10.1.2. *Modificarea coeficienţilor funcţiei obiectiv*

Dacă după rezolvarea problemei iniţiale vectorul coeficienţilor funcţiei obiectiv devine $\tilde{\mathbf{C}}' = \mathbf{C}' + \delta\mathbf{C}'$, atunci soluţia $\overline{\mathbf{X}}^{\mathbf{B}} = \mathbf{B}^{-1}\mathbf{b}$ rămîne neschimbată însă se modifică diferenţele ΔZ_j:

$$\tilde{z}_j - \tilde{c}_j = \tilde{\mathbf{C}}'_{\mathbf{B}} \cdot \mathbf{B}^{-1} \cdot a_j - \tilde{c}_j = \mathbf{C}'_{\mathbf{B}}\mathbf{B}^{-1}a_j - c_j + (\delta\mathbf{C}')_{\mathbf{B}}\mathbf{B}^{-1}a_j - (\delta c_j)$$

Pe baza analizei noilor diferenţe se deosebesc două cazuri:

1) Toate diferenţele $\tilde{z}_j - \tilde{c}_j$ corespunzătoare vectorilor care nu aparţin bazei **B** păstrează semnul pe care l-au avut înainte de modificarea vectorului **C** ($\leqslant 0$ pentru problema de minim şi $\geqslant 0$ pentru problema de maxim), ceea ce înseamnă că şi noua soluţie va fi optimă.

Valoarea funcţiei obiectiv în problema modificată, este:

$$\min(\max)\ \tilde{\mathbf{C}}'\mathbf{X} = \tilde{\mathbf{C}}'\overline{\mathbf{X}}^{\mathbf{B}} = (\mathbf{C}' + \delta\mathbf{C}')_{\mathbf{B}} \cdot \overline{\mathbf{X}}^{\mathbf{B}} = \mathbf{C}'_{\mathbf{B}}\overline{\mathbf{X}}^{\mathbf{B}} + (\delta\mathbf{C}')_{\mathbf{B}}\overline{\mathbf{X}}^{\mathbf{B}}.$$

Din această relaţie, rezultă că funcţia obiectiv ia o altă valoare numai dacă modificarea vectorului **C** afectează componentele corespunzătoare variabilelor de bază.

2) Cel puţin una dintre diferenţele $\tilde{z}_j - \tilde{c}_j$, corespunzătoare unui vector ce nu aparţine bazei, este pozitivă în cazul problemei de minim şi negativă în cazul problemei de maxim. În această situaţie soluţia optimă a problemei iniţiale (nemodificate) nu mai este optimă şi se continuă optimizarea ei prin aplicarea algoritmului simplex primal pornind de la ultimul tabel simplex.

Observaţie: Procedeul indicat la reoptimizare în cazul cînd se modifică vectorul **C**, poate fi aplicat şi în cazul problemelor care au acelaşi sistem de restricţii, dar diferă prin obiectivul urmărit.

Exemplul 2.15. Pe baza problemei de programare liniară, prezentate în exemplul 2.14 ne propunem să vedem dacă soluţia optimă dată în tabelul 2.20 se modifică atunci cînd vectorul coeficienţilor funcţiei obiectiv devine $\tilde{\mathbf{C}}' = (10,\ 9,\rho)$. Introducînd aceste componente în tabelul 2.20

---

* Consumul de materii prime importate este de numai 350 unităţi fizice deoarece în soluţia optimă dată în tabelul T_2 apare variabila $x_6 = 150$ care reprezintă cantitatea nefolosită din această resursă.

Scanned with OKEN Scanner

se constată că diferența $\Delta z_2 = z_2 - c_2 = 1/12$ (vezi tabelul 2.22), deci soluția nu mai este optimă.

Tabelul 2.22

B	c_B	$\bar{X}$	10	9	9	0	0	0	λ	θ_i
			a_1	a_2	a_3	a_4	a_5	a_6	a_7	
a_1	11	325/3	1	0	0	−13/24	−7/24	−5/12	13/24	—
a_2	9	25/3	0	1	0	−1/24	(5/24)	−5/12	1/24	40
a_3	9	25	0	0	1	3/8	1/8	3/4	−3/8	200
z_j		1383,33	10	9	9	−29/12	1/12	−7/6	29/12	—
Δz_j		—	0	0	0	−29/12	1/12	−7/6	29/12	$-\lambda$
a_1	10	120	1	7/5	0	−3/5	0	−1	3/5	
a_5	0	40	0	24/5	0	−1/5	1	−2	1/5	
a_3	9	20	0	−3/5	1	2/5	0	1	−2/5	
z_j		1 380	10	43/5	9	−12/15	0	−1	12/15	
Δz_j		—	0	−2/5	0	−12/5	0	−1	12/15	$-\lambda$

Soluția optimă a acestei probleme s-a obținut prin aplicarea algoritmului simplex primal.

2.10.1.3 *Adăugarea unor vectori coloană în matricea* A

Dacă programul de producție trebuie să includă noi produse care nu au fost luate în considerație la alcătuirea modelului inițial reoptimizarea constă în adaptarea soluției optime aflate pentru problema inițială la problema mărită. Pentru aceasta, va trebui să vedem dacă noii vectori coloană introduși, satisfac criteriul de optimalitate. În acest scop, va trebui să se calculeze componentele vectorului a_l, $l \in \tilde{N}$ ($\tilde{N}$ este mulțimea indicilor vectorilor noi introduși) în baza optimă B, după acesta putem calcula diferențele $z_l - c_l$, $l \in \tilde{N}$.

Pot apărea acum următoarele situații:

1) Condițiile de optimalitate sînt verificate pentru toți vectorii a_l, $l \in \tilde{N}$ și în acest caz, soluția optimă a problemei inițiale rămîne optimă și pentru problema mărită.

2) Există cel puțin un vector a_l, $l \in \tilde{N}$ care nu verifică criteriul de optimalitate și în acest caz se continuă aplicarea algoritmului simplex pentru determinarea soluției optime.

Exemplul 2.16. Presupunem că la problema prezentată în exemplul 2.14 se mai adugă încă două tipuri de utilaje a căror prețuri de achiziție sînt 12 și respectiv 14 unități monetare, iar caracteristicile tehnico-economice incluse în restricțiile problemei formează vectorii $a'_8 = (6, 5, 4)$ și respectiv $a'_9 = (7,4,3)$.

Cu ajutorul tehnicii de reoptimizare descrisă în paragraful precedent se poate verifica dacă includerea în matricea A a vectorilor coloană a_8 și a_9 modifică soluția optimă dată în tabelul 2.20.

Scanned with OKEN Scanner

Folosind relaţia $y_l = \mathbf{B}^{-1} \cdot a_l$ $(l = 8, 9)$ se obţin componentele vectorilor a_8 şi a_9 în bază $\mathbf{B} = (a_1, a_2, a_3)$:

$$\begin{pmatrix} y_{18} & y_{19} \\ y_{28} & y_{29} \\ y_{38} & y_{39} \end{pmatrix} = \begin{pmatrix} 13/24 & -7/24 & -5/12 \\ 1/24 & 5/24 & -5/12 \\ -3/8 & 1/8 & 3/4 \end{pmatrix} \cdot \begin{pmatrix} 6 & 7 \\ 5 & 4 \\ 4 & 3 \end{pmatrix} = \begin{pmatrix} 1/8 & 11/8 \\ -3/8 & -1/8 \\ 11/8 & 1/8 \end{pmatrix}$$

Pe baza acestor componente se calculează diferenţele $z_l - c_l$, obţinîndu-se

$$\Delta z_8 = z_8 - c_8 = \left(10\frac{1}{8} - 7 \cdot \frac{3}{8} + 9\frac{11}{8}\right) - 12 = 11 - 12 = -1$$

$$\Delta z_9 = z_9 - c_9 = \left(10 \cdot \frac{11}{8} - 7 \cdot \frac{1}{8} + 9 \cdot \frac{1}{8}\right) - 14 = 14 - 14 = 0$$

Se constată că vectorului a_8 îi corespunde o diferenţă negativă deci nu este eficientă introducerea lui în bază. Întrucît $\Delta z_9 = 0$ rezultă că noua problemă are două soluţii optime de bază. Pentru a afla această nouă soluţie se ataşează coloana y_9 tabelului 2.20 şi se introduce în bază vectorul a_9 prin aplicarea algoritmului simplex primal (vezi tabelul 2.23). Trebuie precizat că vectorul a_8 nu a fost luat în considerare la reoptimizare deoarece diferenţa Δz_8 respectă criteriul de optimalitate şi pentru baza $\widetilde{\mathbf{B}} = (a_9, a_2, a_3)$.

Tabelul 2.23

B	c_B	$\bar{X}$	10	7	9	0	0	0	λ	14
			a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_9
a_1	10	325/3	1	0	0	−13/24	−7/24	−5/12	13/24	(11/8)
a_2	7	25/3	0	1	0	−1/24	5/24	−5/12	1/24	−1/8
a_3	9	25	0	0	1	3/8	1/8	3/4	−3/8	1/8
z_j		1366,67	10	7	9	−7/3	−1/3	−1/3	7/3	14
Δz_j		—	0	0	0	−7/3	−1/3	−1/3	7/3 − λ	0*
a_9	14	2600/33	8/11	0	0	−13/33	−7/33	−10/33	13/33	1
a_2	7	600/33	1/11	1	0	−1/11	2/11	−5/11	1/11	0
a_3	9	500/33	−1/11	0	1	14/33	5/33	26/33	−14/33	0
z_j		1366,67	10	7	9	−7/3	−1/3	−1/3	7/3	14
Δz_j		—	0*	0	0	−7/3	−1/3	−1/3	7/3 − λ	0

Cele două soluţii de bază optime sînt :

$$\bar{X}^{\mathbf{B}} = \left(\frac{975}{9}, \frac{75}{9}, 25, 0, 0, 0, 0, 0,\right)$$

$$\bar{X}^{\widetilde{\mathbf{B}}} = \left(0, \frac{600}{33}, \frac{500}{33}, 0, 0, 0, 0, \frac{2600}{33}\right)$$

Scanned with OKEN Scanner

Se observă că valoarea funcției obiectiv pentru cele două soluții este aceeași. Orice combinație liniară convexă a soluțiilor $\overline{\mathbf{X}}^{\mathbf{B}}$ și $\overline{\mathbf{X}}^{\tilde{\mathbf{B}}}$ dă o altă soluție optimă nebazică, adică :

$$\overline{\mathbf{X}} = \alpha\,\overline{\mathbf{X}}^{\mathbf{B}} + (1-\alpha)\overline{\mathbf{X}}^{\tilde{\mathbf{B}}}$$

Spre exemplu, pentru $\alpha = 0{,}6$ se obține :

$$\overline{\mathbf{X}} = (65;\ 12{,}101;\ 21{,}066;\ 0;\ 0;\ 0;\ 0;\ 31,\ 515).$$

2.10.1.4. *Modificarea unui vector coloană al matricei* **A**

Vom analiza cazul în care se modifică vectorul coloană a_r din matricea **A**, după ce s-a obținut soluția optimă $\overline{\mathbf{X}}^{\mathbf{B}}$.

Evident, vectorul modificat a_r, poate să aparțină sau nu bazei **B** și în funcție de aceasta avem de considerat două situații.

1) Vectorul a_r nu aparține bazei **B**. În acest caz, verificarea optimalității soluției $\overline{\mathbf{X}}^{\mathbf{B}}$ în noile condiții, se reduce la calcularea diferenței $z_r - c_r$.

2) Vectorul a_r aparține bazei **B**. În acest caz componentele vectorului modificat $\tilde{a}_r$ în baza **B** sînt :

$$y_{ir} = \beta_i \tilde{a}_r$$

unde β_i este vectorul ale cărei componente sînt elementele de pe linia i din matricea $\mathbf{B}^{-1}$.

Însă, $\tilde{a}_r$ poate înlocui în bază pe a_r numai dacă componenta vectorului $\tilde{a}_r$ corespunzătoare lui a_r este diferită de zero, adică

$$y_{rr} = \beta_r \tilde{a}_r \neq 0$$

Dacă această condiție este îndeplinită, va trebui să calculăm inversa bazei **B**, obținută din baza **B** prin înlocuirea vectorului a_r cu vectorul $\tilde{a}_r$.

Pentru aceasta, procedăm în felul următor.

Se calculează vectorul

$$\mathbf{g} = (g_1, g_2, \ldots, g_m) = \mathbf{B}^{-1}\tilde{a}_r$$

și se alcătuiește vectorul

$$\mathbf{g}^* = \left(-\frac{g_1}{g_r}, -\frac{g_2}{g_r}, \ldots, -\frac{g_{r-1}}{g_r}, \frac{1}{g_r}, \frac{g_{r+1}}{g_r}, \ldots, -\frac{g_m}{g_r}\right)$$

Se scrie matricea unitate de ordinul m în care vectorul e_r se înlocuiește prin $\mathbf{g}^*$ și se notează această matrice $\mathbf{E}_r^{(m)}$.

Cu această matrice, obținem inversa căutată.

$$\mathbf{B}^{-1} = \mathbf{E}_r^{(m)}\,\mathbf{B}^{-1}$$

Dispunînd acum de inversa bazei $\tilde{\mathbf{B}}$, se calculează ușor

$$\overline{\mathbf{X}}^{\tilde{\mathbf{B}}} = \tilde{\mathbf{B}}^{-1}.\mathbf{b}$$

Scanned with OKEN Scanner

și diferențele

$$z_j - c_j = C'_{\tilde{B}} \tilde{B}^{-1} a_j - c_j.$$

cu ajutorul cărora testăm optimalitatea acestei soluții. În afară de situațiile întîlnite la analiza optimalității soluției în cazul algoritmului simplex primal și dual, poate apărea următoarea situație:

Vectorul $\overline{X}^{\tilde{B}}$ are componente negative și nu sînt verificate toate condițiile de optim. În aceste cazuri, se poate relua problema de la iterația anterioară intrării vectorului a_r în bază sau se poate înlocui $\tilde{a}_r = a_r + (\delta a_r)$, atașînd variabila x_r vectorului a_r și o nouă variabilă x'_r vectorului δa_r ceea ce conduce la mărirea matricei **A**, cu o coloană, caz pe care l-am analizat mai înainte. Soluția este valabilă și pentru cazul în care vectorul $\tilde{a}_r$ nu îndeplinește condiția necesară pentru a înlocui pe a_r în bază (adică cînd $y_{rr} = \beta_r \tilde{a}_r = 0$).

Exemplul 2.17. Analizînd caracteristicile tehnice ale celor trei utilaje considerate în problema prezentată în exemplul 2.14 s-a ajuns la concluzia că se pot efectua unele îmbunătățiri tehnice utilajului de tipul 2, rezultînd următorul vector al indicatorilor tehnico-economici: $\tilde{a}_2 = (5,5,1)$. Urmează să se determine efectul acestor îmbunătățiri tehnice asupra soluției optime.

În acest scop se calculează componentele vectorului $g = \mathbf{B}^{-1} \tilde{a}_2$:

$$g = \begin{pmatrix} 13/24 & -7/24 & -5/12 \\ 1/24 & 5/24 & -5/12 \\ -3/8 & 1/8 & 3/4 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 5 \\ 1 \end{pmatrix} = \begin{pmatrix} 5/6 \\ 5/6 \\ -1/2 \end{pmatrix}$$

Se constată că $g_2 = \tilde{y}_{22} = 5/6$ deci putem calcula componentele vectorului g*, obținîndu-se $g^* = (-1, 6/5, 3/5)$. Pe baza acestor componente se construiește matricea $\mathbf{E}_2^{(3)}$:

$$E_2^{(3)} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 6/5 & 0 \\ 0 & 3/5 & 1 \end{pmatrix}$$

Rezultă că inversa matricei $\tilde{\mathbf{B}} = (a_1, \tilde{a}_2, a_3)$ se determină din următoarea relație:

$$\tilde{B}^{-1} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 6/5 & 0 \\ 0 & 3/5 & 1 \end{pmatrix} \cdot \begin{pmatrix} 13/24 & -7/24 & -5/12 \\ 1/24 & 5/24 & -5/12 \\ -3/8 & 1/8 & 3/4 \end{pmatrix} = \begin{pmatrix} 1/2 & -1/2 & 0 \\ 1/20 & 1/4 & -1/2 \\ -7/20 & 1/4 & 1/2 \end{pmatrix}$$

Deci prin înlocuirea vectorului a_2 cu $\tilde{a}_2$ se obține o nouă soluție $\overline{X}^{\tilde{B}}$:

$$\overline{X}^{\tilde{B}} = \begin{pmatrix} 1/2 & -1/2 & 0 \\ 1/20 & 1/4 & -1/2 \\ -7/20 & 1/4 & 1/2 \end{pmatrix} \cdot \begin{pmatrix} 700 \\ 500 \\ 300 \end{pmatrix} = \begin{pmatrix} 100 \\ 10 \\ 30 \end{pmatrix}$$

După calcularea componentelor vectorilor a_2 și a_4 în baza $\tilde{\mathbf{B}}$ prin relația $y_j = \tilde{B}^{-1} \cdot a_j$ $(j = 2,4)$ se obțin toate informațiile necesare construirii

Scanned with OKEN Scanner

tabelului simplex 2.24 T_1. Acest tabel a rezultat din tabelul 2.20 după efectuarea înlocuirilor corespunzătoare.

Tabelul 2.24

	B	c_B	$\bar{X}$	10	7	9	0	0*	0	λ	7
				a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_2
T_1	a_1	10	100	1	−1	0	−1/2	−1/2	0	1/2	0
	$\tilde{a}_2$	7	10	0	6/5	0	−1/20	1/4	−1/2	1/20	1
	a_3	9	30	0	3/5	1	−7/20	1/4	(1/2)	−7/20	0
	z_j		1 340	10	19/5	9	−11/5	−1	1	11/5	7
	Δz_j		—	0	−16/5	0	−11/5	−1	1	11/5−λ	0
T_2	a_1	10	100	1	−1	0	−1/2	−1/2	0	1/2	0
	$\tilde{a}_2$	7	40	0	9/5	1	3/10	1/2	0	−3/10	1
	a_6	0	60	0	6/5	2	7/10	1/2	1	−7/10	0
	z_j		1 280	10	13/5	7	−29/10	−3/2	0	29/10	7
	Δz_j			0	−22/5	−2	−29/10	−3/2	0	29/10−λ	0

Se observă că în tabelul 2.24 T_1 apare diferența $\Delta z_6 = 1$ și de aceea se introduce în bază vectorul a_6. Noua soluție optimă din tabelul 2.24 T_2 are două efecte economice importante:

— reducerea eforturilor valutare pentru achiziționarea mijloacelor de muncă;

— reducerea consumului de materii prime importate (întrucît $x_6 = 60$ reprezintă o economie de materii prime importate).

2.10.1.5. *Adăugarea unor restricții*

Considerăm cazul general în care se adaugă sistemului inițial de restricții încă r relații de tipul:

$$\sum_{j=1}^{n} a_{m+p,j}\, x_j + x_{n+m+p} = b_{m+p} \quad (p = 1,2,\ldots,r)$$

Înlocuind variabilele x_k $(k \in \mathbf{B})$ în relațiile de mai sus, obținem r funcții liniare de variabilele x care nu aparțin bazei. Termenii liberi din aceste funcții liniare notate b_{m+p} pot fi numere pozitive, nule sau negative.

Completînd vectorul $\bar{X}$ cu componentele $\bar{x}_{n+m+p} = b_{m+p}$ $(1 \leqslant p \leqslant r)$ se obține vectorul $\bar{X}$, care este o soluție de bază pentru sistemul format din ecuațiile:

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+i} = b_i \quad (1 \leqslant i \leqslant m)$$

$$\sum_{j=1}^{n} a_{m+p,j}\, x_j + x_{n+m+p} = b_{m+p} \quad (1 \leqslant p \leqslant r)$$

Scanned with OKEN Scanner

Vom considera următoarele cazuri:

1) Toate valorile $b_{m+p} \geqslant 0$, $(1 \leqslant p \leqslant r)$ și toate variabilele x_{n+m+p} sînt variabile abatere. În acest caz $\overline{\overline{\mathbf{X}}}$ este o soluție optimă pentru problema completată cu noile condiții.

Într-adevăr, $\overline{\overline{\mathbf{X}}} \geqslant 0$ iar condițiile de optimalitate sînt satisfăcute deoarece diferențele $z_j - c_j$ nu se modifică ca urmare a faptului că vectorul $\mathbf{C_B}$ se completează cu componente nule corespunzătoare variabilelor abatere x_{n+m+p} ce intră în funcția obiectiv.

2) Toate valorile $b_{m+p} \geqslant 0$ dar cel puțin una din variabilele x_{n+m+p} este variabilă artificială. În acest caz, soluția $\overline{\overline{\mathbf{X}}}$ include variabile artificiale și se poate folosi pentru excluderea lor din bază, metoda penalizărilor sau faza I din metoda celor două faze.

3) Unele din valorile b_{m+p} sînt nenegative iar $\overline{\overline{\mathbf{X}}}$ este o soluție dual realizabilă. Pentru optimizarea ei se folosește mai departe algoritmul simplex dual.

Să vedem acum, cum se poate face eliminarea variabilelor x_k $(k \in \mathbf{B})$ din restricțiile adăugate, utilizînd numai tabelul simplex optim al problemei inițiale, adică vrem să calculăm inversa bazei curente completate cu variabilele abatere și artificiale ale noilor restricții, deoarece cunoașterea acestei matrici inverse a bazei este suficientă pentru determinarea atît a constantelor cît și pentru continuarea iterațiilor. Pentru a pune în evidență modul de lucru, vom proceda în felul următor.

Notăm cu

$$\widetilde{\mathbf{B}} = \left(\begin{array}{c|c} \mathbf{B} & \mathbf{0} \\ \hline \mathbf{B}_1 & \mathbf{I}^{(r)} \end{array}\right) \text{ și } \widetilde{\mathbf{B}}^{-1} = \left(\begin{array}{c|c} \mathbf{P} & \mathbf{0} \\ \hline \mathbf{Q} & \mathbf{R} \end{array}\right)$$

unde $\mathbf{B}_1$ este o submatrice de tipul $(r, n+m)$ formată din coeficienții $a_{m+p,j}$ $(p = 1,2,\ldots,r;\ j = 1,2,\ldots,n+m)$, iar $\mathbf{I}^{(r)}$ este o submatrice unitate de tipul (r,r).

Avînd în vedere faptul că:

$$\widetilde{\mathbf{B}} \cdot \widetilde{\mathbf{B}}^{-1} = \left(\begin{array}{c|c} \mathbf{I}^{(m)} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I}^{(r)} \end{array}\right)$$

rezultă:

$$\widetilde{\mathbf{B}}^{-1} = \left(\begin{array}{c|c} \mathbf{B}^{-1} & \mathbf{0} \\ \hline -\mathbf{B}_1\,\mathbf{B}^{-1} & \mathbf{I}^{(r)} \end{array}\right)$$

Deoarece matricea $\mathbf{B}^{-1}$ este cunoscută din tabelul simplex final al problemei nemodificate, se poate calcula imediat $\widetilde{\mathbf{B}}^{-1}$.

Scanned with OKEN Scanner

Exemplul 2.18. Presupunem că la problema de programare liniară prezentată în exemplul 2.14 se adaugă următoarele două restricţii :

$$0{,}2x_1 + 0{,}6x_2 + 0{,}6x_3 \geqslant 0$$
$$x_3 \leqslant 20$$

Prima restricţie s-a obţinut din relaţia :

$$[5x_1p - 0{,}56\,(5x_1p)] + [4x_2p - 0{,}5\,(4x_2p)] + [5x_3p - 0{,}52\,(5x_3p)] \geqslant$$
$$\geqslant \frac{1}{d_n}(10x_1 + 7x_2 + 9x_3)$$

unde p reprezintă preţul unitar de vînzare a produsului fabricat cu cele trei tipuri de utilaje, indicatorii 0,56 ; 0,50 şi respectiv 0,52 reprezintă ponderea costului producţiei fabricate într-un an pe fiecare tip de utilaj, iar d_n este durata normată de amortizare a fondurilor fixe achiziţionate. Avînd în vedere că termenii din partea stîngă a inegalităţii cuantifică venitul net ce se obţine în cursul unui an, în ipoteza folosirii complete a capacităţii de producţie, iar termenul din partea dreaptă reprezintă valoarea utilajelor ce urmează a fi amortizată într-un an, rezultă că această restricţie va asigura includerea cu prioritate în soluţia optimă a utilajelor la care se realizează un venit net mai mare.

Pentru $p = 1$ şi $d_n = 5$ se obţine, după trecerea tuturor termenilor în membrul stîng, prima dintre cele două restricţii adăugate.

În vederea reoptimizării este necesar să se calculeze matricea $\widetilde{\mathbf{B}}^{-1}$ folosind inversa matricii $\mathbf{B} = (a_1, a_2, a_3)$ dată în tabelul 2.20 şi noile restricţii adăugate.

Întrucît :

$$-\mathbf{B}_1\mathbf{B}^{-1} = \begin{pmatrix} 0{,}2 & 0{,}6 & 0{,}6 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 13/24 & -7/24 & -5/12 \\ 1/24 & 5/24 & -5/12 \\ -3/8 & 1/8 & 3/4 \end{pmatrix} = \begin{pmatrix} 11/120 & -17/120 & -7/60 \\ 3/8 & -1/8 & -3/4 \end{pmatrix}$$

rezultă că vectorul $\overline{\overline{\mathbf{X}}}^{\tilde{\mathbf{B}}} = \widetilde{\mathbf{B}}^{-1} \cdot \overline{\overline{\mathbf{b}}}'$ va avea componentele :

$$\overline{\overline{\mathbf{X}}}^{\tilde{\mathbf{B}}} = \begin{pmatrix} 13/24 & -7/24 & -5/12 & \cdot & 0 & 0 \\ 1/24 & 5/24 & -5/12 & \cdot & 0 & 0 \\ -3/8 & 1/8 & 3/4 & \cdot & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 11/120 & -17/120 & -7/60 & \cdot & 1 & 0 \\ 3/8 & -1/8 & -3/4 & \cdot & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 700 \\ 500 \\ 300 \\ \cdots \\ 0 \\ 20 \end{pmatrix} = \begin{pmatrix} 325/3 \\ 25/3 \\ 25 \\ \cdots \\ -125/3 \\ -5 \end{pmatrix}$$

Deoarece vectorul $\overline{\overline{\mathbf{X}}}^{\tilde{\mathbf{B}}}$ are componente negative şi $\Delta z_j^{\tilde{\mathbf{B}}} \leqslant 0$, se poate aplica algoritmul simplex dual pentru optimizarea acesteia. Calculele necesare optimizării se dau în tabelele 2.25.

Scanned with OKEN Scanner

Tabelul 2.25

B	c_B	$\bar{X}$	10	7	9	0	0	0	λ	0	λ	0
			a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
a_1	10	325/3	1	0	0	$-13/24$	$-\frac{7}{24}$	$-\frac{5}{12}$	$\frac{13}{24}$	0	0	0
a_2	7	25/3	0	1	0	$-1/24$	$5/24$	$-5/12$	$1/24$	0	0	0
a_3	9	25	0	0	1	$3/8$	$1/8$	$3/4$	$-3/8$	0	0	0
a_9	λ	$-125/3$	0	0	0	$-\frac{11}{120}$	$-\frac{17}{120}$	$-\frac{7}{60}$	$\frac{11}{120}$	-1	1	0
a_{10}	0	-5	0	0	0	$-3/8$	$-1/8$	$-3/4$	$3/8$	0	0	1
Δz_j		—	0	0	0	$-\frac{11}{120}\lambda$ $-7/3$	$-\frac{17}{120}\lambda$ $-1/3$	$-\frac{7}{60}\lambda$ $-1/3$	$-\frac{109}{120}\lambda$ $+7/3$	$-\lambda$	0	0
a_1	10	325/3	1	0	0	$-13/24$	$-7/24$	$-5/12$	$13/24$	0	0	0
a_2	7	25/3	0	1	0	$-1/24$	$5/24$	$-5/12$	$1/24$	0	0	0
a_3	9	25	0	0	1	$3/8$	$1/8$	$3/4$	$-3/8$	0	0	0
a_8	0	125/3	0	0	0	$11/120$	$17/120$	$7/60$	$-11/120$	1	-1	0
a_{10}	0	-5	0	0	0	$-3/8$	$-1/8$	$-3/4$	$3/8$	0	0	1
Δz_j		—	0	0	0	$-7/3$	$-1/3$	$-1/3$	$7/3-\lambda$	0	$-\lambda$	0
a_1	10	1000/9	1	0	0	$-1/3$	$-2/9$	0	$1/3$	0	0	$-5/9$
a_2	7	100/9	0	1	0	$1/6$	$5/18$	0	$-1/6$	0	0	$-5/9$
a_3	9	20	0	0	1	0	0	0	0	0	0	1
a_8	0	368/9	0	0	0	$1/30$	$11/90$	0	$-1/30$	1	-1	$7/45$
a_6	0	20/3	0	0	0	$1/2$	$1/6$	1	$-1/2$	0	0	$-4/3$
Δz_j			0	0	0	$-13/6$	$-5/18$	0	$-\frac{13}{6}-\lambda$	0	$-\lambda$	$-4/9$

Scanned with OKEN Scanner

2.10.2. Programarea parametrică

Modelarea diferitelor fenomene şi procese economice cu ajutorul programării parametrice îşi găseşte aplicabilitate în rezolvarea problemelor pe care le ridică variaţia unor elemente considerate constante în modelele de programare liniară. Necesitatea tratării „dinamice" a mărimilor prestabilite ale modelelor de programare liniară îşi are suportul în situaţiile reale întîlnite în practica economică.

Astfel, componentele vectorului termenilor liberi pot înregistra modificări faţă de valoarea estimată b_i datorită unor cauze interne sau externe sistemului economic analizat. De exemplu cantitatea disponibilă de fier, cupru, zinc etc. depinde de conţinutul de substanţă minerală utilă din minereul ce urmează a fi prelucrat; capacitatea disponibilă a mijloacelor de muncă este funcţie de calitatea utilajului ce urmează a fi instalat şi de fondul de investiţii alocat; cererea de bunuri de consum poate fi exprimată ca o funcţie de ritmul de creştere al populaţiei şi a retribuţiei etc.

Şi în cazul parametrizării coeficienţilor funcţiei obiectiv, la exprimarea acestora ca funcţii de anumiţi parametri trebuie să se ţină seama de natura economică a funcţiei obiectiv. Astfel, dacă coeficienţii c_j reprezintă costul unitar de producţie atunci costul total de producţie $\sum_j c_j x_j$ poate fi exprimat ca funcţie liniară sau neliniară de un parametru α de care depinde variaţia coeficienţilor c_j; dacă coeficienţii c_j reprezintă beneficiul unitar obţinut prin realizarea unei unităţi din produsul j, atunci beneficiul total $\sum_j c_j x_j$ depinde de costul de producţie efectiv realizat (şi de preţul de realizare în cazul cînd produsul se livrează la export).

Cu ajutorul programării parametrice se pot analiza şi aspectele legate de variaţia coeficienţilor matricei **A**. De exemplu la stabilirea programului de lucru al laminoarelor trebuie să se studieze variaţia coeficienţilor de utilizare a laminoarelor în funcţie de temperatura la care se face laminarea şi de calitatea oţelului laminat. În acest caz matricea **A** va conţine o coloană ale cărei elemente vor fi funcţii de cei doi parametri care pot lua valori în anumite intervale.

Rezultă că determinarea programului optim presupune analiza soluţiilor posibile corespunzătoare unor valori ale parametrilor incluşi în model şi alegerea variantei optime.

Studiul unor astfel de probleme se face cu ajutorul modelelor pe care ni le oferă „programarea parametrică" utilizînd şi noi denumirea cunoscută în literatura de specialitate.

Caracteristic programării parametrice este faptul că unii dintre coeficienţi („constantele") modelului matematic sînt funcţii liniare sau neliniare de unul sau mai mulţi parametri.

Avînd în vedere cele arătate mai înainte rezultă că principalele cazuri de programare parametrică sînt următoarele:

— parametrizarea vectorului termenilor liberi;
— parametrizarea vectorului coeficienţilor funcţiei obiectiv;
— parametrizarea unui vector coloană sau linie din matricea **A**.

Rezolvarea problemelor de programare parametrică comportă parcurgerea a două etape:

Scanned with OKEN Scanner

1. determinarea soluției optime pentru o valoare determinată a parametrului (sau a unui sistem determinant de valori dacă este vorba de mai mulți parametri);

2. odată determinată soluția optimă pentru o valoare fixată, a parametrului, se face studiul sensibilității soluției optime la variația parametrului.

Acest studiu constă în stabilirea intervalelor sau domeniilor de optimalitate a unei soluții, adică a mulțimii valorilor parametrilor pentru care soluția își păstrează caracterul de optimalitate.

Din punct de vedere practic interesează cazurile în care parametrii α_i ce intervin în diferitele modele sînt numere reale cuprinse în intervale finite $\alpha_i \in [\beta_i, \delta_i]$ care prin schimbări convenabile de origine se pot transforma în intervale de forma $[0, \gamma_i]$.

Dificultățile întîmpinate la rezolvarea unei probleme de programare parametrică sînt mult mai mari ca cele întîmpinate la reoptimizare. În această lucrare ne vom ocupa numai de cazul parametrizării vectorului $\mathbf{b}$ și $\mathbf{C}$.

2.10.2.1. *Vectorul* $\mathbf{b}$ *este funcție liniară de un parametru*

Considerăm problema de programare liniară dată sub forma standard

$$\max. [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$

$$\mathbf{AX} = \mathbf{b}(\alpha)$$

$$\mathbf{X} \geqslant 0$$

$$\text{unde } \mathbf{b}(\alpha) = \mathbf{b}^{(0)} + \alpha\mathbf{b}^{(1)}, \ \alpha \in \mathbf{R}$$

Vom presupune că se cunoaște o soluție optimă a problemei de programare liniară parametrizată, corespunzătoare unei valori α date a parametrului. Se vede imediat că putem considera $\alpha = 0$.

Atunci, soluția problemei prezentate mai sus, asociată bazei $\mathbf{B}$ este:

$$\overline{\mathbf{X}}_{\mathbf{B}}(\alpha) = \mathbf{B}^{-1}\mathbf{b}\ (\alpha) = \mathbf{B}^{-1}(\mathbf{b}^{(0)} + \alpha\mathbf{b}^{(1)}) = \overline{\mathbf{X}}_{\mathbf{B}}^{(0)} + \alpha\overline{\mathbf{X}}_{\mathbf{B}}^{(1)}$$

unde am notat

$$\overline{\mathbf{X}}_{\mathbf{B}}^{(0)} = \mathbf{B}^{-1}\mathbf{b}^{(0)}; \ \overline{\mathbf{X}}_{\mathbf{B}}^{(1)} = \mathbf{B}^{-1}\mathbf{b}^{(1)}$$

Pentru ca această soluție să fie optimă, ea trebuie să fie realizabilă, adică componentele sale să fie nenegative:

$$x_i(\alpha) = x_i^{(0)} + \alpha x_i^{(1)} \geqslant 0, \ i \in \mathbf{B} \tag{2.28}$$

Întrucît $\overline{\mathbf{X}}_{\mathbf{B}}^{(0)}$ este o soluție optimă pentru $\alpha = 0$, componentele sale sînt nenegative, de aceea relația (2.28) este satisfăcută pentru $\alpha \in [\underline{\alpha}, \overline{\alpha}]$ unde

$$\underline{\alpha} = \begin{cases} \max\left(-\dfrac{x_i^{(0)}}{x_i^{(1)}}\right) \text{dacă } x_i^{(1)} > 0 \text{ pentru cel puțin un indice } i \in \mathbf{B} \\ -\infty \text{ dacă } x_i^{(1)} \leqslant 0 \text{ pentru toate valorile lui } i. \end{cases}$$

Scanned with OKEN Scanner

$$\bar{\alpha} = \begin{cases} \min\left(-\dfrac{x_i^{(0)}}{x_i^{(1)}}\right) \text{ dacă } x_i^{(1)} < 0 \text{ pentru cel puțin un indice } i \in \mathbf{B}. \\ +\infty \text{ dacă } x_i^{(1)} \geqslant 0 \text{ pentru orice indice } i \in \mathbf{B}. \end{cases}$$

Dacă analizăm comportarea soluției pentru $\alpha > \bar{\alpha}$ se constată că pentru $\alpha = \bar{\alpha} + \varepsilon$ ($\varepsilon > 0$, arbitrar ales) cel puțin o componentă a vectorului $\bar{X}_{\mathbf{B}}(\bar{\alpha} + \varepsilon)$ devine negativă, însă soluția rămîne dual realizabilă deoarece condițiile de optimalitate rămîn neschimbate.

Presupunînd că $\bar{\alpha}$ este unic determinat, adică

$$\bar{\alpha} = \min\left(-\frac{x_i^{(0)}}{x_{(i)}^{(1)}}\right) = -\frac{x_l^{(0)}}{x_l^{(1)}}$$

rezultă că $x_l(\bar{\alpha} + \varepsilon) = x_l^{(0)} + (\bar{\alpha} + \varepsilon) x_l^{(1)} = x_l^{(0)} + \bar{\alpha} x_l^{(1)} + \varepsilon x_l^{(1)} = \varepsilon x_l^{(1)} < 0$

Prin aplicarea algoritmului simplex dual se va elimina din bază variabila x_l (dacă există componente $y_{lj} < 0$) și în locul ei va intra variabila x_k. Deci, după o iterație * se obține o nouă bază optimă $\mathbf{B}_1$ căreia îi corespunde intervalul de optimalitate $[\bar{\alpha}, \alpha_1]$. Trebuie precizat că pentru $\alpha = \bar{\alpha}$ ambele baze sînt optime.

Se poate demonstra că $\alpha_1 > \bar{\alpha}$ și atunci datorită faptului că numărul vectorilor coloană din matricea $\mathbf{A}$ este finit, rezultă că după un număr finit de iterații vom împărți intervalul $[\bar{\alpha}, \infty]$ în subintervale de optimalitate corespunzătoare unor baze determinate. În acest scop plecăm de la soluția asociată bazei $\mathbf{B}_1$:

$$\bar{X}_{\mathbf{B}_1}(\bar{\alpha}) = \mathbf{B}_1^{-1}[\mathbf{b}^{(0)} + \bar{\alpha}\mathbf{b}^{(1)}] = \mathbf{B}_1^{-1}\mathbf{b}^{(0)} + \bar{\alpha}\mathbf{B}_1^{-1}\mathbf{b}^{(1)}$$

componentele acestui vector sînt:

$$\bar{x}_i(\bar{\alpha}) = \bar{x}_i^{(0)} + \bar{\alpha}\bar{x}_i^{(1)} \geqslant 0 \quad i \in [\mathbf{B} - (l)]$$

$$\bar{x}_k(\bar{\alpha}) = 0$$

Dacă $\alpha = \bar{\alpha} + \varepsilon$ obținem

$$\bar{x}_i(\bar{\alpha} + \varepsilon) = \bar{x}_i^{(0)} + \bar{\alpha}\bar{x}_i^{(1)} + \varepsilon\bar{x}_i^{(1)} = \bar{x}_i(\bar{\alpha}) + \varepsilon\bar{x}_i^{(1)} \quad (i \in \mathbf{B} - \{l\})$$

$$\bar{x}_k(\bar{\alpha} + \varepsilon) = \frac{1}{y_{lk}}\bar{x}_l^{(1)}$$

Dacă vectorul $\bar{X}_{\mathbf{B}_1}^{(1)} = \mathbf{B}_1^{-1}\mathbf{b}^{(1)}$ are toate componentele nenegative, atunci $\mathbf{B}_1$ este baza optimă pentru orice $\varepsilon > 0$ adică pentru $\alpha \in [\bar{\alpha}, \infty]$.

Dacă componenta $\bar{x}_l^{(1)} < 0$, a vectorului $\bar{X}_{\mathbf{B}_1}^{(1)}$, atunci $\mathbf{B}_1$ nu este optimă pentru $\varepsilon > 0$; problema nu admite soluții realizabile pentru $\alpha > \bar{\alpha}$.

* Dacă se obține aceeași valoare minimă $\bar{\alpha}$ pentru mai mulți indici $i \in I_1$ atunci devin negative toate componentele $x_l(\bar{\alpha} + \varepsilon)$ pentru $l \in I_1$ fiind necesare mai multe iterații pentru obținerea soluției optime.

Scanned with OKEN Scanner

Dacă $\bar{x}_i^{(1)}$ este pozitiv, dar există componenta $\bar{x}_i^{(1)} < 0$ pentru $i \in [\mathbf{B} - \{l\} + \{k\}]$ atunci condiţiile de nenegativitate a componentelor soluţiei $\overline{\mathbf{X}}_{\mathbf{B}_1}$ sînt îndeplinite pentru

$$\varepsilon = \min \left(\frac{\bar{x}_i^{(o)}(\bar{\alpha})}{\bar{x}_i^{(1)}(\bar{\alpha})} \right)$$

Întrucît $\bar{x}_i(\bar{\alpha}) > 0$ pentru soluţii nedegenerate, rezultă că $\varepsilon > 0$ şi $\mathbf{B}_1$ este o bază optimă pentru $\alpha \in [\bar{\alpha}, \alpha_1]$ unde $\alpha_1 > \bar{\alpha}$.

Tratamentul pe care l-am aplicat bazei $\mathbf{B}$ îl aplicăm acum bazei $\mathbf{B}_1$ şi obţinem o nouă bază $\mathbf{B}_2$ şi corespunzător ei o valoare α_2 şi tot aşa mai departe vom determina succesiunea de baze $\mathbf{B}_1, \mathbf{B}_2 \ldots \mathbf{B}_p$ şi succesiunea corespunzătoare de valori critice $\alpha_1, \alpha_2, \ldots, \alpha_p$, limite ale intervalelor de optimalitate corespunzătoare.

Cum am procedat pentru $\alpha > \bar{\alpha}$ vom proceda şi cu $\alpha < \underline{\alpha}$ şi vom determina succesiunea de baze optime $\mathbf{B}_1', \mathbf{B}_2', \ldots, \mathbf{B}_q'$ cu valorile critice corespunzătoare ale parametrului α $(\alpha_1', \alpha_2', \ldots, \alpha_q')$.

Din aceste consideraţii, rezultă că axa reală am împărţit-o în intervale cărora le corespunde cîte o bază optimă. Bazele asociate intervalelor care au cîte o extremitate comună, diferă prin cîte un singur vector.

Pentru valorile critice ale parametrului — extremităţi ale unui interval — problema admite o infinitate de soluţii determinate ca o *combinaţie liniară convexă* de cele două soluţii optime corespunzătoare bazelor ce diferă printr-un singur vector.

Exemplul 2.19. Prin rezolvarea modelului matematic corespunzător problemei prezentate în exemplul 2.14 se obţine o soluţie optimă care nu ţine seama de variaţia posibilă a componentelor vectorului termenilor liberi. Pentru a avea o imagine mai completă asupra variantelor de dezvoltare s-a analizat posibilitatea livrării la export a produsului fabricat cu cele trei tipuri de utilaje, a creşterii capacităţilor de producţie la întreprinderile ce fabrică materia primă de tipul $\mathbf{M}_1$ şi a extinderii importului de materii prime de tipul $\mathbf{M}_2$, rezultînd următorul vector al termenilor liberi*

$$\mathbf{b}(\alpha) = \begin{pmatrix} 700 + 12\alpha \\ 500 + 20\alpha \\ 300 + 10\alpha \end{pmatrix}$$

Se calculează $\overline{\mathbf{X}}_{\mathbf{B}}(\alpha) = \mathbf{B}^{-1} \cdot \mathbf{b}(\alpha)$ obţinîndu-se:

$$\overline{\mathbf{X}}_{\mathbf{B}}(\alpha) = \begin{pmatrix} 13/24 & -7/24 & -5/12 \\ 1/24 & 5/24 & -5/12 \\ -3/8 & 1/8 & 3/4 \end{pmatrix} \cdot \begin{pmatrix} 700 + 12\alpha \\ 500 + 20\alpha \\ 300 + 10\alpha \end{pmatrix} = \begin{pmatrix} 325/3 \\ 25/3 \\ 25 \end{pmatrix} + \begin{pmatrix} -7/2\alpha \\ 1/2\alpha \\ 11/2\alpha \end{pmatrix}$$

---

* Pentru prima componentă, α reprezintă numărul de contracte cu o cantitate de 12 unităţi fizice ce se pot realiza; la cea de a doua componentă, α reprezintă numărul capacităţilor de producţie a 20 unităţi fizice ce se vor instala la întreprinderile de fabrică materii prime $\mathbf{M}_2$, iar la cea de a treia componentă α, reprezintă numărul de contracte cu o cantitate de 10 unităţi fizice ce se pot realiza pentru importul materiei prime $\mathbf{M}_2$.

Scanned with OKEN Scanner

De aici se determină valorile critice ale parametrului α:

$$\underline{\alpha} = \max\left(-\frac{25}{3}\cdot 2, -25\cdot\frac{2}{11}\right) = -\frac{50}{11}$$

$$\overline{\alpha} = \frac{325}{3}\cdot\frac{2}{7} = \frac{650}{21}$$

Deci pentru valori ale parametrului α cuprinse în intervalul

$$\left[-\frac{50}{11}, \frac{650}{21}\right]$$ baza $\mathbf{B} = (a_1, a_2, a_3)$ rămînă optimă.

Dacă $\alpha < -50/11$ atunci devine negativă variabila x_3 şi de aceea vectorul a_3 trebuie eliminat din bază folosind algoritmul simplex dual. Însă componentele $y_{3j} > 0$, $(j = 4,5,6)$ şi ca urmare problema nu are soluţie pentru $\alpha < \underline{\alpha}$ (vezi tabelul 2.20)

În cazul cînd $\alpha > 650/21$ variabila x_1 va deveni negativă şi pentru eliminarea acesteia din bază se aplică algoritmul simplex dual. În acest scop s-au folosit tabelele 2.26.

Tabelul 2.26.

B	c_B	$\bar{X}(\alpha)$	10	7	9	0	0	0	λ
			a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_1	10	$\frac{325}{3} - \frac{7}{2}\alpha$	1	0	0	−13/24	−7/24	−5/12	13/24
a_2	7	$\frac{25}{3} + \frac{1}{2}\alpha$	0	1	0	−1/24	5/24	−5/12	1/24
a_3	9	$25 + \frac{11}{2}\alpha$	0	0	1	3/8	1/8	3/4	−3/8
Δz_j			0	0	0	−7/3	−1/3	−1/3	$\frac{7}{3} - \lambda$
a_6	0	$-260 + \frac{42}{5}\alpha$	−4/5	0	0	13/10	7/10	1	−13/10
a_2	7	$-100 + 4\alpha$	−1	1	0	1/2	1/2	0	−1/2
a_3	9	$220 - \frac{4}{5}\alpha$	9/5	0	1	−3/5	−2/5	0	3/5
Δz_j			−4/5	0	0	−19/10	−1/10	0	$\frac{19}{10} - \lambda$

Scanned with OKEN Scanner

Din ultimul tabel simplex rezultă $\alpha_1 = 220 \cdot \frac{5}{4} = 275$, iar baza $B_1 = (a_6, a_2, a_3)$ este optimă în intervalul $[650/21, 275]$. Dacă $\alpha > 275$, atunci va ieşi din bază vectorul a_3 şi în locul lui va intra a_5.

Tabelul 2.27

B	C_B	$\bar{X}(\alpha)$	10	7	9	0	0	0	λ
			a_1	a_2	a_3	a_4	a_5	a_6	a_7
a_6	0	$125 + 7\alpha$	47/20	0	7/4	1/4	0	1	$-1/4$
a_2	7	$175 + 3\alpha$	5/4	1	5/4	$-1/4$	0	0	1/4
a_5	0	$-550 + 2\alpha$	$-9/2$	0	$-5/2$	3/2	1	0	$-3/2$
Δz_j		—	$-5/4$	0	$-1/4$	$-7/4$	0	0	$7/4 - \lambda$

Întrucît $\bar{x}_i^{(1)} > 0 (i = 1, 2, 3)$, rezultă că $\alpha_2 = \infty$ iar baza $B_2 = (a_6, a_2, a_5)$ este optimă în intervalul $[275, \infty]$.

2.10.2.2 *Parametrizarea vectorului coeficienţilor funcţiei obiectiv.*

Modelul matematic al unei probleme de programare liniară în care coeficienţii funcţiei obiectiv sînt funcţii liniare de un parametru α are forma :

$$\min (\max) [Z(x)] = C'(\alpha) \cdot X$$

în condiţiile

$$AX = b$$

$$X \geqslant 0$$

unde $C(\alpha) = C^{(0)} + \alpha C^{(1)}$ cu $\alpha \in R$.

Studiul mulţimii soluţiilor optime necesită cunoaşterea soluţiei optime corespunzătoare unei valori particulare a parametrului $\alpha (\alpha = \alpha_0$ sau $\alpha = 0)$. Dacă $\alpha = 0$ atunci se rezolvă problema de mai sus pentru $C(\alpha) = C^{(0)}$, obţinîndu-se baza optimă B corespunzătoare acestei valori.

După cum este cunoscut, modificarea componentelor vectorului C are efect numai asupra diferenţelor $z_j - c_j$, acestea avînd forma :

$$z_j(\alpha) - c_j(\alpha) = C'_B(\alpha) \cdot B^{-1} \cdot a_j - c_j(\alpha) = (C_B^{(0)} + \alpha C_B^{(1)})' \cdot B^{-1} \cdot a_j - (c_j^{(0)} + \alpha c_j^{(1)}) = C_B^{(0)} \cdot B^{-1} \cdot a_j - c_j^{(0)} + \alpha (C_B^{(1)} \cdot B^{-1} a_j - c_j^{(1)}) = (z_j^{(0)} - c_j^{(0)}) + \alpha (z_j^{(1)} - c_j^{(1)}).$$

Dacă ne referim la o problemă de maxim, analiza acestor diferenţe pune în evidenţă următoarele situaţii :

1) $z_j^{(0)} - c_j^{(0)} \geqslant 0$ pentru orice $j \in J_s$ şi ca urmare s-a găsit soluţia optimă pentru $\alpha = 0$.

Scanned with OKEN Scanner

2) Există $z_s^{(o)} - c_s^{(o)} < 0$ pentru o submulțime de vectori $a_s \in \mathbf{J_S}$ și toate componentele y_{is}, $i \in \mathbf{J_B}$ ale acestor vectori în baza **B** sînt negative. Atunci problema nu are optim finit pentru $\alpha = 0$.

Dacă sistemul de inecuații $(z_s^{(o)} - c_s^{(o)}) + \alpha (z_s^{(1)} - c_s^{(1)}) > 0$ pentru $s \in I_1 \subset \subset \mathbf{J}_S$ nu este compatibil, sau dacă o diferență $z_s^{(1)} - c_s^{(1)} = 0$ pentru $s \in \mathbf{J_B}$, atunci problema nu are optim finit pentru nici o valoare a parametrului α.

Dacă sistemul $(z_j^{(o)} - c_j^{(o)}) + \alpha (z_j^{(1)} - c_j^{(1)}) > 0$ este compatibil, se determină cea mai mică valoare a parametrului α care satisface toate inecuațiile. Baza **B** este optimă pentru $\alpha = \alpha_0$ și problema nu are optim finit pentru $\alpha < \alpha_0$.

Să arătăm că sistemul

$$z_s^{(o)} - c_s^{(o)} + \alpha (z_s^{(1)} - c_s^{(1)}) > 0$$

este incompatibil, dacă și numai dacă nu toate diferențele $z_s^{(1)} - c_s^{(1)}$ sînt de același semn. Într-adevăr, să presupunem că există un indice s_1 pentru care $z_{s_1}^{(1)} - c_{s_1}^{(1)} < 0$ și condiția

$$z_{s_1}^{(o)} - c_{s_1}^{(o)} + \alpha (z_{s_1}^{(1)} - c_{s_1}^{(1)}) > 0$$

este satisfăcută, atunci :

$$\alpha < \frac{z_{s_1}^{(o)} - c_{s_1}^{(o)}}{-(z_{s_1}^{(1)} - c_{s_1}^{(1)})} < 0 \quad \text{deoarece}$$

$$-(z_{s_1}^{(1)} - c_{s_1}^{(1)}) > 0 \text{ și } (z_{s_1}^{(o)} - c_{s_1}^{(o)}) < 0.$$

Dacă există și un indice $s_2 \in I_2 \subset \mathbf{J_S}$ pentru care $(z_{s_2}^{(1)} - c_{s_2}^{(1)}) > 0$ și este satisfăcută condiția

$$(z_{s_2}^{(o)} - c_{s_2}^{(o)}) + \alpha (z_{s_2}^{(1)} - c_{s_2}^{(1)}) > 0$$

atunci

$$\alpha > \frac{-(z_{s_2}^{(o)} - c_{s_2}^{(o)})}{z_{s_2}^{(1)} - c_{s_2}^{(1)}} > 0,$$

condiții care nu pot fi simultan îndeplinite, ceea ce demonstrează afirmația făcută.

Așadar, în problemele de programare parametrică în care vectorul **C** este funcție liniară, de un singur parametru, se pot alcătui tabelele corespunzătoare iterațiilor simplex, ținînd seama numai de componentele vectorului $\mathbf{C}^{(0)}$ și după un număr finit cu iterații, se ajunge la una din următoarele situații :

(*i*) Se găsește o bază **B** optimă pentru o valoare determinată a parametrului α.

(*ii*) Se constată că problema nu are soluție optimă pentru nici o valoare a parametrului α.

Cazul (*ii*) nu prezintă interes practic mai departe și drept urmare ne ocupăm în continuare de cazul (*i*).

Așadar, presupunem că am determinat baza **B** corespunzătoare valorii $\alpha = \alpha_0$ a parametrului. Fără a restrînge generalitatea problemei vom pre-

Scanned with OKEN Scanner

supune că $\alpha = 0$, ceea ce se poate face totdeauna printr-o schimbare convenabilă a originii.

Această bază, rămîne optimă pentru toate valorile α pentru care:

$$(z_j^{(o)} - c_j^{(o)}) + \alpha (z_j^{(1)} - c_j^{(1)}) \geqslant 0 \quad j \in J_S$$

Cum **B** este optimă pentru $\alpha = 0$, rezultă că

$$(z_j^{(o)} - c_j^{(o)}) \geqslant 0, \; j \in J_S .$$

Deci putem determina $[\underline{\alpha}, \bar{\alpha}]$, astfel încît pentru $\alpha \in [\underline{\alpha}, \bar{\alpha}]$ să avem **B** bază optimă, unde

$$\underline{\alpha} = \begin{cases} \max \left(- \dfrac{z_j^{(o)} - c_j^{(o)}}{z_j^{(1)} - c_j^{(1)}} \right) & \text{dacă } (z_j^{(o)} - c_j^{(o)}) > 0 \text{ pentru cel puțin un indice } j \in J_S \\ -\infty \text{ dacă } (z_j^{(1)} - c_j^{(1)}) \leqslant 0 \text{ pentru toți } j \in J_S \end{cases}$$

și

$$\bar{\alpha} = \begin{cases} \min \left(- \dfrac{z_j^{(o)} - c_j^{(o)}}{z_j^{(1)} - c_j^{(1)}} \right) & \text{dacă există } x_j^{(1)} - c_j^{(1)} < 0 \text{ pentru cel puțin un indice } j \in J_S \\ \infty \text{ dacă } (x_j^{(1)} - c_j^{(1)}) \geqslant 0 \text{ pentru toți } j \in J_S \end{cases} \qquad (2.29)$$

Prezintă interes cazul cînd cel puțin una dintre valorile critice este finită.

Dacă $\bar{\alpha}$ are valoarea finită, atunci se pot determina o serie de subintervale ale intervalului $[\bar{\alpha}, \infty)$ și bazele optime corespunzătoare. În cazul în care $\bar{\alpha}$ este unic determinat din relația (2.29) adică:

$$\bar{\alpha} = \min \left(- \frac{z_j^{(o)} - c_j^{(o)}}{z_j^{(1)} - c_j^{(1)}} \right) = - \frac{z_k^{(o)} - c_k^{(o)}}{z_k^{(1)} - c_k^{(1)}}$$

vectorul a_k va intra în noua bază deoarece $z_k^{(1)} - c_k^{(1)} < 0$ și pentru un $\varepsilon > 0$ obținem:

$$z_k(\bar{\alpha} + \varepsilon) - c_k(\bar{\alpha} + \varepsilon) = [(z_k^{(o)} - c_k^{(o)}) + (\bar{\alpha} + \varepsilon)(z_k^{(1)} - c_k^{(1)})] < 0$$

Conform algoritmului simplex primal, vectorul a_k va intra în noua bază $\mathbf{B}_1$ (dacă există $y_{lk} > 0$) în locul vectorului a_l stabilit pe baza criteriului de ieșire din bază. Calculînd diferențele $\Delta z_j(\bar{\alpha} + \varepsilon)$ corespunzătoare bazei $\mathbf{B}_1$ obținem:

$$z_k(\bar{\alpha} + \varepsilon) - c_k(\bar{\alpha} + \varepsilon) = 0$$

$$z_j(\bar{\alpha} + \varepsilon) - c_j(\bar{\alpha} + \varepsilon) = (z_j^{(o)} - c_j^{(o)}) + (\bar{\alpha} + \varepsilon)(z_j^{(1)} - c_j^{(1)}) -$$

$$- \frac{y_{lj}}{y_{lk}} \cdot \varepsilon (z_k^{(1)} - c_k^{(1)}) = (z_j^{(o)} - c_j^{(o)}) + \bar{\alpha}(z_j^{(1)} - c_j^{(1)}) +$$

$$+ \varepsilon \left[(z_j^{(1)} - c_j^{(1)}) - \frac{y_{lj}}{y_{lk}} (z_k^{(1)} - c_k^{(1)}) \right].$$

Scanned with OKEN Scanner

Dacă $(z_j^{(1)} - c_j^{(1)}) - \frac{y_{ij}}{y_{ik}}(z_k^{(1)} - c_k^{(1)}) \geqslant 0$ pentru toţi $j \in \mathbf{J_S} - \{k\}$, rezultă că:

$z_j(\bar{\alpha} + \varepsilon) - c_j(\bar{\alpha} + \varepsilon) \geqslant 0$ pentru orice ε şi deci noua bază $\mathbf{B}_1$ este optimă pentru toate valorile lui α.

Dacă însă $(z_j^{(1)} - c_j^{(1)}) - \frac{y_{ij}}{y_{ik}}(z_k^{(1)} - c_k^{(1)}) < 0$ pentru o mulţime de indici $j \in \mathbf{J_S} - \{k\}$ se determină valoarea

$$\varepsilon = \min \left[- \frac{(z_j^{(0)} - c_j^{(0)}) + \bar{\alpha}(z_j^{(1)} - c_j^{(1)})}{(z_j^{(1)} - c_j^{(1)}) - \frac{y_{ij}}{y_{ik}}(z_k^{(1)} - c_k^{(1)})} \right]$$

pentru $\left[(z_j^{(1)} - c_j^{(1)}) - \frac{y_{ij}}{y_{ik}}(z_k^{(1)} - c_k^{(1)}) \right] < 0.$

Se vede că $\varepsilon > 0$.

De aici, rezultă că baza $\mathbf{B}_1$ este optimă pentru

$$\alpha \in [\bar{\alpha}, \alpha_1] \text{ unde } \alpha_1 = \bar{\alpha} + \varepsilon.$$

Dacă α_1 este infinit, problema s-a încheiat iar dacă nu, aplicînd mai departe acelaşi procedeu se determină succesiunea de baze $\mathbf{B}_2, \mathbf{B}_3, \ldots, \mathbf{B}_p$ optime pentru $\alpha \in [\alpha_{h-1}\alpha_h]$, $h = 2,3,\ldots,p$ (unde α_p poate fi ∞) pînă se acoperă tot intervalul de variaţie al parametrului α.

Un raţionament analog, ne permite să construim succesiunea de bază $\tilde{\mathbf{B}}_1, \tilde{\mathbf{B}}_2, \ldots, \tilde{\mathbf{B}}_q$ optime pentru $\alpha \in [\alpha'_r, \alpha'_{r-1}]$, $r = 2,3,\ldots,q$, cu $\alpha'_1 = \underline{\alpha}$ unde α'_q poate fi $-\infty$.

Valorile $\alpha_1, \alpha_2, \ldots, \alpha_p$ şi $\alpha'_1, \alpha'_2 \ldots, \alpha'_q$ se numesc valori critice ale parametrului şi ele împart mulţimea valorilor acestui parametru în submulţimi caracterizate de faptul că fiecărei dintre aceste submulţimi îi este asociată o bază optimă.

Fiecărei valori critice îi corespund două baze optime, deci o infinitate de soluţii optime obţinute ca combinaţii liniare de cele două soluţii optime.

În cazul în care $\bar{\alpha}$ nu este unic determinat, diferenţele $z_j(\bar{\alpha}) - c_j(\bar{\alpha})$ se anulează pentru o submulţime de indici $\mathbf{S}_1 \in \mathbf{S}$ iar diferenţele corespunzătoare vor fi negative pentru $\alpha = \alpha_1 + \varepsilon$ ($\varepsilon > 0$) şi vor fi necesare mai multe iteraţii pentru obţinerea unei baze optime.

Procedeul de obţinere a lor este analog cu cazul cînd $\underline{\alpha}$ este unic determinat cu singura deosebire că bazele corespunzătoare valorilor parametrului $\alpha > \bar{\alpha}$ şi $\alpha < \underline{\alpha}$ diferă prin mai mult decît un vector.

Scanned with OKEN Scanner

2.11. Probleme de programare liniară de tip transport

2.11.1 Problema transporturilor

2.11.1.1. *Formularea problemei și proprietăți generale*

În această secțiune vom analiza o problemă particulară de programare liniară, frecvent întîlnită în aplicații și cunoscută sub numele de „problema transporturilor". Din punct de vedere economic această problemă se enunță astfel :

În centrele de origine A_i $(1 \leqslant i \leqslant m)$ se află un produs care trebuie transportat la centrele de destinație B_j $(1 \leqslant j \leqslant n-1)$. Cunoscîndu-se cantitățile disponibile a_i și necesarurile b_j precum și costurile unitare de transport c_{ij} de la centrul A_i la centrul B_j (vezi tabelul 2.28), se cere să se stabilească planul de transport astfel încît cheltuielile totale de transport să fie minime.

Tabelul 2.28

B_j / A_i	B_1	B_2 ... B_{n-1}	Disponibil
A_1	c_{11}	$c_{12} \ldots c_{1n-1}$	a_1
A_2	c_{21}	$c_{22} \ldots c_{2n-1}$	a_2
.	.	. .	.
.	.	. .	.
.	.	. .	.
A_m	c_{m1}	$c_{m2} \ldots c_{mn-1}$	a_m
Necesar	b_1	b_2 ... b_{n-1}	

Dacă notăm cu x_{ij} cantitatea transportată de la A_i la B_j, atunci modelul matematic al problemei va fi :

$$\min\ [Z(x)] = \sum_{i=1}^{m} \sum_{j=1}^{n-1} c_{ij} x_{ij}$$

în condițiile :

$$\sum_{j=1}^{n-1} x_{ij} \leqslant a_i \qquad (1 \leqslant i \leqslant m)$$

$$\sum_{i=1}^{m} x_{ij} \geqslant b_j \qquad (1 \leqslant j \leqslant n-1)$$

$$x_{ij} \geqslant 0$$

$$\sum_{i=1}^{m} a_i \geqslant \sum_{j=1}^{n-1} b_j,\ a_i \geqslant 0,\ b_j \geqslant 0,$$

Scanned with OKEN Scanner

Se demonstrează uşor că această problemă se poate înlocui cu o problemă echivalentă în care toate restricţiile sînt ecuaţii.

Astfel, datorită relaţiei $\sum_{i=1}^{m} a_i \geqslant \sum_{j=1}^{n-1} b_j$, cel de-al doilea grup de inecuaţii se verifică cu egalitate pentru orice soluţie de bază.

Totodată putem introduce variabilele ecart x_{in} $(1 \leqslant i \leqslant m)$ în fiecare din primele m inecuaţii astfel încît să devină ecuaţii.

Variabilele ecart sînt legate prin relaţia :

$$\sum_{i=1}^{m} x_{in} = \sum_{i=1}^{m} a_i - \sum_{j=1}^{n-1} b_j = b_n.$$

Această transformare poate fi interpretată ca adăugarea unui centru de destinaţie fictiv, în care se transportă — cu costuri unitare nule — surplusul existent în fiecare din cele m centre de desfacere.

Ca rezultat se obţine modelul matematic :

$$\min [Z(x)] = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad (1 \leqslant i \leqslant m)$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (1 \leqslant j \leqslant n)$$

$$x_{ij} \geqslant 0$$

în ipotezele $a_i \geqslant 0$, $b_j \geqslant 0$, $c_{ij} \geqslant 0$, $c_{in} = 0$, $\sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j$.

În această formă spunem că avem o problemă de transport echilibrată şi toate consideraţiile de natură teoretică le vom face pe o astfel de problemă.

În aplicaţiile practice se întîlnesc şi situaţii care impun elaborarea unui plan optim de transport pentru cazul $\sum_{i=1}^{m} a_i \leqslant \sum_{j=1}^{n} b_j$. Această inegalitate face ca primul grup de restricţii al modelului iniţial să se verifice cu egalitate pentru orice soluţie de bază, iar al doilea grup va conţine variabile ecart cărora le corespund costuri nule în funcţia obiectiv. Suma variabilelor ecart se interpretează astfel :

$$\sum_{j=1}^{n} x_{m+1,j} = \sum_{j=1}^{n} b_j - \sum_{i=1}^{m} a_i = a_{m+1}$$

Scanned with OKEN Scanner

Prin introducerea unui producător (depozit) fictiv A_{m+1} se obține tot o problemă de transport echilibrată.

Pentru problema de transport astfel formulată este adevărată următoarea teoremă:

Teoremă. O problemă de transport echilibrată admite totdeauna programe; orice program este mărginit dacă sînt mărginite valorile a_i și b_j.

Demonstrație. Se constată imediat că

$$x_{ij} = \frac{a_i b_j}{\sum_{i=1}^{n} a_i}$$

este un program și de asemenea că $x_{ij} \leqslant \min(a_i, b_j)$.

Deci valorile x_{ij} sînt finite dacă sînt finite mărimile a_i și b_j.

Scrierea matricială a problemei de transport ne dă posibilitatea să constatăm că este de forma:

$$\min [\mathbf{Z}(x)] = \mathbf{C}'\mathbf{X}$$
$$\mathbf{AX} = \mathbf{b}$$
$$\mathbf{X} \geqslant 0$$

unde matricea coeficienților tehnologici este de forma:

$$\mathbf{A} = \begin{pmatrix} 1\ 1\ \ldots\ 1 & 0\ 0\ \ldots\ 0 & \ldots & 0\ 0\ \ldots\ 0 \\ 0\ 0\ \ldots\ 0 & 1\ 1\ \ldots\ 1 & \ldots & 0\ 0\ \ldots\ 0 \\ \hline 0\ 0\ \ldots\ 0 & 0\ 0\ \ldots\ 0 & \ldots & 1\ 1\ \ldots\ 1 \\ 1\ 0\ \ldots\ 0 & 1\ 0\ \ldots\ 0 & \ldots & 1\ 0\ \ldots\ 0 \\ 0\ 1\ \ldots\ 0 & 0\ 1\ \ldots\ 0 & \ldots & 0\ 1\ \ldots\ 0 \\ \hline 0\ 0\ \ldots\ 1 & 0\ 0\ \ldots\ 1 & \ldots & 0\ 0\ \ldots\ 1 \end{pmatrix} \begin{matrix} \} m \\ \\ \\ \} n \\ \\ \\ \end{matrix}$$

$$\underbrace{}_{n} \quad \underbrace{}_{n} \quad \underbrace{}_{n}$$

de m ori n

iar

$$\mathbf{C}' = (c_{11}c_{12},\ldots,c_{1n};\ c_{21},c_{22},\ldots,c_{2n};\ \ldots;\ c_{m1},\ c_{m2},\ldots,c_{mn})$$
$$\mathbf{X}' = (x_{11},x_{12},\ldots,x_{1n};\ x_{21},x_{22},\ldots,x_{2n};\ldots;\ x_{m1}\ldots,x_{mn})$$
$$\mathbf{b}' = (a_1,\ a_2,\ldots,a_m;\ b_1,\ b_2,\ldots,b_n)$$

Adesea matricea $\mathbf{A}$ o scriem sub forma

$$\mathbf{A} = (a_{11},\ a_{12},\ldots,a_{1n};\ a_{21},\ldots,a_{2n};\ldots;a_{m1},\ldots,a_{mn})$$

unde

$$a_{kl} = a_k + e_{m+l},$$

e_i fiind vectorul unitar cu $m + n$ componente.

Teorema de mai jos precizează rangul matricii $\mathbf{A}$.

Teoremă. Matricea $\mathbf{A}$ are rangul $m + n - 1$.

Demonstrație. Deoarece rang $\mathbf{A} \leqslant \min(m + n, m \cdot n)$ și $m+n \leqslant m \cdot n$, dacă $m, n \geqslant 2$ rezultă rang $\mathbf{A} \leqslant m + n$.

Scanned with OKEN Scanner

Dar rang $\mathbf{A} < m+n$ deoarece suma primelor m linii este egală cu suma ultimelor n linii în matricea **A**. (Acest lucru este o consecinţă a faptului că $\sum_i a_i = \sum_j b_j$).

Din **A** se poate pune în evidenţă o matrice pătratică de ordinul $m+n-1$ cu determinantul diferit de zero. Într-adevăr să suprimăm linia $m+1$, apoi să luăm în matricea rămasă coloanele $1, n+1, 2n+1, \ldots, mn-n+1, 2, 3, 4, \ldots, n$. Elementele ce formează diagonala principală a acestei matrici sînt egale cu 1, iar cele situate sub diagonală sînt egale cu zero. Determinantul acestei matrici fiind 1, rezultă că rang $\mathbf{A} = n + m - 1$.

În cele ce urmează ne vom referi la tabelul problemei de transport care prin definiţie va fi un tablou dreptunghiular cu m linii şi n coloane şi care va conţine $m \cdot n$ căsuţe. Liniile tabelului corespund celor m depozite iar coloanele corespund celor n beneficiari. În fiecare căsuţă (i, j) a tabloului T, situată la intersecţia liniei i cu coloana j, figurează următoarele elemente :

— costul unitar de transport c_{ij}

— variabila x_{ij}.

Tabloul este bordat cu o coloană marginală care cuprinde disponibilităţile a_i şi cu o linie marginală ce conţine necesarul b_j.

Tabelul 2.29

x_{11} c_{11}	x_{12} c_{12}	. . .	x_{1n} c_{1n}	a_1
x_{21} c_{21}	x_{22} c_{22}	. . .	x_{2n} c_{2n}	a_2
⋮	⋮	⋮	⋮	⋮
x_{m1} c_{m1}	x_{m2} c_{m2}	. . .	x_{mn} c_{mn}	a_m
b_1	b_2	. . .	b_n	

2.11.1.2. *Algoritmul simplificat pentru determinarea soluţiei optime a problemei de transport.*

Să pornim de la problema de transport echilibrată şi duala ei :

$$\min\ [\mathbf{Z}(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = a_i \quad (i = 1,2,\ldots,m)$$

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = 1,2,\ldots,n)$$

$$x_{ij} \geqslant 0 \quad (i = 1,2,\ldots,m;\ j = 1,2,\ldots,n)$$

$$\max\ [\mathbf{G}(u,v)] = \sum_{i=1}^{m} a_i u_i + \sum_{j=1}^{n} b_j v_j$$

u_i oarecare

v_j oarecare

$$u_i + v_j \leqslant c_{ij}$$

Scanned with OKEN Scanner

Atunci conform teoremei slabe a ecarturilor complementare (vezi secţiunea 2.8.1) ansamblul de valori ale variabilelor x_{ij}, u_i, $v_j (i = 1, 2, \ldots, m; j = 1,2,\ldots,n)$ constituie programe optimale ale cuplului de probleme de transport dacă şi numai dacă

$$\sum_{j=1}^{n} x_{ij} = a_i \qquad i = 1,2,\ldots,m$$

$$\sum_{i=1}^{m} x_{ij} = b_j \qquad j = 1, 2,\ldots,n$$

$$x_{ij} \geqslant 0$$

$$c_{ij} - u_i - v_j \geqslant 0$$

$$x_{ij}(c_{ij} - u_i - v_j) = 0$$

Ultima relaţie exprimă faptul că dacă $x_{ij} \neq 0$ rezultă $(c_{ij} - u_i - v_j = 0)$, deci $u_i + v_j = c_{ij}$.

Dar după cum ştim $x_{ij} \neq 0$ numai dacă sînt variabile de bază. (Dacă programul este nedegenerat numai variabilele bazice sînt nenule.)

Notăm cu $\mathbf{M} = \{1,2,\ldots,m\}$, $\mathbf{N} = \{1,2,\ldots,n\}$ şi $\mathbf{I} \subset \mathbf{M} \times \mathbf{N}$ mulţimea indicilor (i, j) pentru care x_{ij} sînt variabile de bază.

Atunci, fiind dat un program de bază şi mulţimea de valori u_i, v_j ce verifică $m + n - 1$ ecuaţii cu $m + n$ necunoscute :

$$u_i + v_j = c_{ij} \qquad (i,j) \in \mathbf{I}$$

programul de bază este optimal dacă şi numai dacă

$$u_i + v_j - c_{ij} \leqslant 0, \qquad (i,j) \in \mathbf{M} \times \mathbf{N}$$

Referitor la sistemul de ecuaţii $u_i + v_j = c_{ij}$, $(i, j) \in \mathbf{I}$ avem următorul rezultat :

Teoremă : Sistemul de $m + n - 1$ ecuaţii liniare cu $m + n$ necunoscute $u_i + v_j = c_{ij}$, $(i, j) \in \mathbf{I}$ are o infinitate de soluţii ce depind liniar de un parametru. Dacă (u_i^0, v_j^0) este o soluţie particulară şi k o constantă arbitrară, atunci orice soluţie este de forma

$$u_i = u_i^0 + k, \quad v_j = v_j^0 - k$$

Demonstraţie. Să considerăm tabelul T de transport. Vom demonstra prin inducţie după liniile tabelului T.

Dacă $m = 1$, n numărul coloanelor arbitrar, atunci sistemul de ecuaţii se va scrie :

$$u_1 + v_j = c_{1j} \quad (j = 1, 2,\ldots,n)$$

Scanned with OKEN Scanner

Soluția generală a acestui sistem este :

$$u_1 = k$$

$$v_j = c_{1j} - k \quad (j = 1,2,\ldots,n)$$

deci teorema este adevărată pentru $m = 1$.

Presupunem că teorema este adevărată pentru tabelul T_{m-1} (cu $m-1$ linii și n coloane) și să considerăm tabelul T_m.

Dar în acest tabel există cel puțin două linii (sau coloane) care conțin numai o singură variabilă de bază ; fără a restrînge generalitatea putem presupune că ultima linie din T_m are această proprietate, adică conține o singură variabilă bazică x_{mj} (pe coloana j). Suprimînd această linie rămîne un tabel T_{m-1} cu $m+n-2$ variabile de bază. Prin ipoteză (inducție) pentru T_{m-1} este adevărat că :

$$u_i = u_i^0 + k \qquad (i = 1,2,\ldots,m-1)$$

$$v_j = v_j^0 - k \qquad (j = 1,2,\ldots,n)$$

Sistemul corespunzător tabelului T_m conține o ecuație în plus

$$u_m + v_j = c_{mj}$$

Înlocuind în această ecuație pe v_j cu valoarea definită mai înainte se obține :

$$u_m = c_{mj} - (v_j^0 - k) = \underbrace{(c_{mj} - v_j^0)}_{u_m^0} + k$$

Aceasta demonstrează complet teorema.

De aici rezultă acum imediat teorema de mai jos.

Teoremă. Dacă $(u_i^0,\ v_j^0)$ este o soluție a sistemului $u_i + v_j = c_{ij}$, $(i,j) \in \mathbf{I}$ corespunzătoare unei baze $\mathbf{B}$ și dacă se pune

$$z_{ij} = \mathbf{c}_B' \mathbf{B}^{-1} a_{ij}$$

atunci

$$z_{ij} - c_{ij} = u_i^0 + v_j^0 - c_{ij}$$

Demonstrația este imediată dacă se ține seama de forma generală a programului dual asociat unei baze determinate.

Pașii algoritmului simplificat pentru obținerea soluției optime a problemei de transport sînt :

a) Se consideră tabelul T asociat problemei de transport și se determină un program inițial de bază care se înscrie în acest tabel.

b) Notînd cu $\mathbf{I}$ mulțimea indicilor $(i,\ j)$ corespunzători variabilelor de bază, rezultă sistemul :

$$u_i + v_j = c_{ij}, \quad (i,j) \in \mathbf{I}.$$

Scanned with OKEN Scanner

Fixîndu-se arbitrar valoarea uneia dintre variabile (de exemplu $u_1 = 0$), sistemul astfel obținut se rezolvă imediat din aproape în aproape. Notînd cu $\bar{u}_i$, $\bar{v}_j$ soluția obținută prin rezolvarea acestui sistem, se calculează expresia $\bar{u}_i + \bar{v}_j - c_{ij}$ pentru orice $(i, j) \notin \mathbf{I}$.

Dacă această expresie este negativă pentru orice $(i, j) \in \mathbf{I}$, programul este minimal. În cazul cînd expresia este pozitivă pentru cel puțin un cuplu (i, j) se determină

$$\bar{u}_k + \bar{v}_l - c_{kl} = \max_{(i,j) \in \mathbf{M} \times \mathbf{N}} (u_i + v_j - c_{ij})$$

și apoi se stabilește ciclul corespunzător cuplului (k, l).

Acest ciclu este constituit din segmente care pun în evidență legătura dintre variabila x_{kl} (ce urmează a intra în bază) și unele variabile ale soluției de bază alese după regula : două variabile succesive din ciclu trebuie să fie situate pe aceeași linie sau coloană. Ca rezultat se obține un „contur poligonal" cu număr par de vîrfuri ocupate de variabile x_{ij} $[(i, j) \in \mathbf{I}]$ și variabila x_{kl}.

c) Se marchează într-un mod oarecare căsuțele ocupate de rang par în succesiunea ciclului cînd acesta este parcurs într-un sens determinat începînd cu căsuța (k,l) care se numerotează cu 1. În căsuțele marcate se caută variabila de valoare minimă.

d) Se scade valoarea minimă găsită din valorile situate în căsuțele de rang par și se adaugă la valorile situate la căsuțele de rang impar. Mulțimea variabilelor de indici $(i, j) \in \mathbf{I} - \{p,q\} + \{k,l\}$, unde (p, q) sînt indicii corespunzători variabilei cu valoarea minimă din ciclu, formează o nouă bază pentru care se reia algoritmul de la pasul b) și se repetă procedeul pînă ce toate diferențele $u_j + v_j - c_{ij} \leqslant 0$, $(i,j) \in \mathbf{M} \times \mathbf{N}$ cînd am obținut soluția optimă.

Observație. Dacă prin procedeul de scădere a valorii minime în căsuțele notate par în ciclu s-au obținut mai multe variabile de valoare zero atunci una dintre ele iese din bază, iar celelalte vor face parte din bază cu valoarea zero, ceea ce înseamnă că vom avea o soluție degenerată.

2.11.1.3. *Obținerea unui program inițial de bază*

Am văzut că algoritmul de determinare a soluției optime presupune ca prim pas existența unei soluții inițiale de bază.

De aceea vom arăta cum se obține o astfel de soluție, avînd în vedere particularitatea problemelor de transport ca problemă de programare liniară dată sub forma standard (probleme echilibrate de transport).

Metoda generală de obținere a unei soluții inițiale, de bază pornește de la determinarea lui

$$\bar{x}_{ij} = \min (a_i, b_j)$$

apoi se înlocuiește a_i și b_j prin $a_i - \bar{x}_{ij}$ și $b_j - \bar{x}_{ij}$ după care se suprimă linia i dacă $\bar{x}_{ij} = a_i$ sau coloana j dacă $\bar{x}_{ij} = b_j$ ceea ce ne dă posibilitatea să obținem un tabel redus, pentru care se repetă procedeul pînă ce sînt satisfăcute toate cererile.

Scanned with OKEN Scanner

Variante ale metodei generale.

a) Metoda colțului de Nord-Vest.

Se începe cu valoarea $\bar{x}_{11} = \min(a_1, b_1)$ și se continuă folosind regula descrisă mai înainte. Caracteristic pentru această metodă este faptul că la fiecare etapă se introduce în bază variabila situată în colțul de N—V al tabloului inițial sau rămas.

Procedeul se pretează pentru rezolvarea problemei pe calculator.

b) Metoda elementului minim pe linie.

Se determină mai întîi $c_{1l} = \min(c_{1j})$ și apoi $\bar{x}_{1l} = \min(a_1, b_l)$

Se înlocuiește a_1 și b_l cu $a_1 - \bar{x}_{1l}$ și respectiv $b_l - \bar{x}_{1l}$ după care se suprimă linia *1*, sau coloana l și se repetă procedeul cu tabelul rămas pînă sînt satisfăcute toate cererile.

c) Metoda minimului pe coloană.

Se caută mai întîi $c_{k1} = \min_i(c_{i1})$ apoi se determină $\bar{x}_{k1} = \min(a_k, b_1)$ și se înlocuiește a_k și b_1 prin $a_k - \bar{x}_{k1}$ și $b_1 - \bar{x}_{k1}$ după care se suprimă linia k sau coloana *1* și se continuă procedeul cu tabelul rămas pînă se ajunge să fie satisfăcute toate cererile.

d) Metoda costului minim din tabel.

Se caută $c_{kl} = \min_{i,j}(c_{ij})$ și apoi se determină $\bar{x}_{kl} = \min(a_k, b_l)$ și se înlocuiește a_k și b_l prin $a_k - \bar{x}_{kl}$ și respectiv $b_l - \bar{x}_{kl}$ după care se suprimă linia k sau coloana l și se obține un tabel cu o linie sau o coloană mai puțin. Se repetă procedeul pînă se ajunge ca toate cererile să fie satisfăcute.

e) Metoda diferențelor maxime.

Se determină pe fiecare linie și fiecare coloană diferența dintre costul imediat superior costului minim și costul minim. Dintre diferențele astfel calculate se alege valoarea maximă. Costul minim de pe linia sau coloana corespunzătoare diferenței maxime indică variabila ce intră în bază. Presupunînd că va intra în bază variabila x_{lk}, rezultă că $x_{lk} = \min(a_l, b_k)$.

Procedeul se repetă pentru tabelul obținut prin suprimarea liniei l sau a coloanei k.

2.11.1.4. *Degenerarea în problemele de transport; Convergența algoritmului pentru obținerea soluției-optime*

Ca orice problemă de programare liniară, problema transporturilor poate să aibă programe de bază degenerate. În acest caz avem o teoremă care caracterizează degenerarea în problemele de transport.

Teoremă. O problemă de transport are programe de bază degenerate atunci și numai atunci cînd există $\mathbf{M}' \subset \mathbf{M}$ și $\mathbf{N}' \subset \mathbf{N}$ astfel încît

$$\sum_{i \in \mathbf{M}'} a_i = \sum_{j \in \mathbf{N}'} b_j$$

Demonstrație. Să presupunem că problema de transport admite un program degenerat. Putem considera, fără a restrînge generalitatea, că $m \leqslant n$. Atunci există în soluția de bază cel puțin o coloană care are în tabelul T numai o căsuță ocupată.

Scanned with OKEN Scanner

Într-adevăr dacă fiecare coloană are cel puţin două căsuţe ocupate, înseamnă că numărul total al acestor căsuţe este mai mare sau egal cu $2n$. Cum numărul componentelor bazice strict pozitive este mai mic sau egal cu $m + n - 1$, rezultă $2n \leqslant m + n + 1$ sau $n \leqslant m-1$ care contrazice ipoteza $m \leqslant n$.

Fie acum $x_{ij} = b_j$ singurul element diferit de zero din coloana j. Suprimînd coloana j vom pune $a'_i = a_i - b_j$ şi obţinem un nou tabel de transport cu n' coloane, unde $n' = n-1 \geqslant m$ sau $n' = n - 1 < m$. Dacă $n' < m$, conform raţionamentului făcut va exista o linie în tabel cu o singură căsuţă ocupată. Fie această căsuţă cea care cuprinde pe x_{kl}.

Dacă $a'_k = 0$, rezultă $k = i$, deci $a'_i = a_i - b_j = 0$ şi afirmaţia teoremei rezultă dacă luăm $\mathbf{M}' = \{i\}$, $\mathbf{N}' = \{j\}$.

Dacă $a'_k \neq 0$ se suprimă linia k şi cantităţile necesare se vor modifica astfel :

$$b'_j = \begin{cases} b_l - a_k & (k \neq i) \\ b_l - (a_i - b_j) & (k = i) \end{cases}$$

La un moment dat se ajunge la $x_{pq} = 0$ care se găseşte în linia p şi coloana q.

Cantităţile disponibile respectiv necesare vor fi

$$a_p - \sum b_r + \sum a_s ; \; b_q - \sum a_s + \sum b_r$$

unde indicii r şi s iau valorile indicilor coloanelor, respectiv liniilor tăiate.

Dacă x_{pq} se află singură pe linie, prima sumă este zero iar r nu ia valoarea q. Analog cînd x_{pq} se află singură pe coloană şi deci rezultă afirmaţia.

Reciproc să presupunem că avem egalitatea de mai sus. Atunci renumerotînd convenabil ecuaţiile şi variabilele (coloanele), putem face ca primele k linii să fie formate cu linii din $\mathbf{M}'$ şi primele l coloane cu cele din $\mathbf{N}'$.

Pentru acest nou tabel vom putea scrie :

$$\sum_{i=1}^{k} a_i = \sum_{j=1}^{l} b_j$$

Deci şi

$$\sum_{i=k+1}^{m} a_i = \sum_{j=l+1}^{n} b_j$$

De aici rezultă că soluţia de bază obţinută prin metoda colţului de N—V va avea

$$(k + l - 1) + (m - k + n - l - 1) = m + n - 2$$

componente nenule şi deci soluţia este degenerată.

Scanned with OKEN Scanner

Se poate însă înlătura degenerarea chiar din etapa de pornire printr-o tehnică simplă de înlocuire a datelor după cum urmează :

$$a_i' = a_i + \varepsilon \qquad (i = 1,2,\ldots,m)$$
$$b_j' = b_j \qquad (j = 1,2,\ldots,n-1)$$
$$b_n' = b_n + m\varepsilon$$

ε fiind o cantitate pozitivă aleasă astfel încît dacă precizia valorilor a_i, b_j este de ordinul 10^{-p} va trebui să satisfacă condiția

$$2m\varepsilon < 10^{-p}$$

Această condiție garantează că nici o sumă parțială de a_i nu va putea fi egală cu o sumă parțială de b_j.

În final pentru a obține soluția problemei considerate se face $\varepsilon = 0$.

Procedeul acesta de perturbare are avantajul că poate fi ușor utilizat în rezolvarea problemelor pe calculator.

Practic se poate renunța la acest procedeu, mulțumindu-ne doar să atribuim valori $\varepsilon_1, \varepsilon_2, \ldots$, diferite și arbitrar de mici, variabilelor ce vor intra succesiv în bază cu valoarea zero.

Exemplul 2.20. O întreprindere de construcții-montaj are în subordine 5 șantiere pentru care aprovizionarea cu ciment se face de la 3 depozite. Disponibilul de ciment al depozitelor este de 680 tone și este prevăzut să acopere necesarul și al altor șantiere ce se vor deschide în viitor.

În tabelul de mai jos sînt trecute disponibilitățile de ciment ale depozitelor, necesitățile șantierelor și distanțele în km dintre depozite și șantiere.

1. Să se determine planul optim de transport și să se stabilească unde trebuie să se păstreze rezerva de 100 tone de ciment.

2. Presupunînd că depozitul D_2 nu mai funcționează, să se determine planul optim de transport și să se precizeze cum se va repartiza deficitul de ciment.

Soluție : 1) Avem o problemă de transport neechilibrată pe care o vom echilibra introducînd un șantier fictiv S_6 (vezi tabelul 2.31 ; distanțele de la depozitele D_i la șantierul S_6 le luăm nule).

Determinăm o soluție inițială de bază utilizînd metoda elementului minim din tabel și o trecem în tabelul 2.32 al soluției inițiale de bază.

Tabelul 2.30

Depozite \ Șantiere	S_1	S_2	S_3	S_4	S_5	Disponibilul de ciment (tone)
D_1	14	8	11	6	12	210
D_2	6	18	15	10	2	200
D_3	12	10	5	11	6	270
Necesarul de ciment (tone)	150	180	50	120	80	680 / 580

Tabelul 2.31.

Depozite \ Şantiere	S_1	S_2	S_3	S_4	S_5	S_6	Disponibil
D_1	14	8	11	6	12	0	210
D_2	6	18	15	10	2	0	200
D_3	12	10	5	11	6	0	270
Necesar	150	180	50	120	80	100	680

Tabelul 2.32

D_i \ S_j	S_1	S_2	S_3	S_4	S_5	S_6	Disponibil
D_1				110 →		→ 100	210
D_2	120			↑	80	↓	200
D_3	30	180	50	10 ←		*	270
Necesar	150	180	50	120	80	100	680

Valoarea funcţiei obiectiv exprimată în tone-km este $\bar{z}^{(0)} = 110 \cdot 6 + 100 \cdot 0 + 120 \cdot 6 + 80 \cdot 2 + 30 \cdot 12 + 180 \cdot 10 + 50 \times 5 + 10 \cdot 11 = 4060$ tone-km.

Testăm acum optimalitatea soluţiei aflate. Pentru aceasta rezolvăm sistemul :

$$u_1 + v_4 = 6 \qquad u_3 + v_1 = 12$$
$$u_1 + v_6 = 0 \qquad u_3 + v_2 = 10$$
$$u_2 + v_1 = 6 \qquad u_3 + v_3 = 5$$
$$u_2 + v_5 = 2 \qquad u_3 + v_4 = 11$$

apoi dacă $\bar{u}_i$ şi $\bar{v}_j$ este soluţia sistemului, calculăm diferenţele $\bar{u}_i + \bar{v}_j - d_{ij}$. Dacă toate aceste diferenţe sînt „$\leqslant 0$" soluţia va fi optimă ; dacă nu, se alege max $(\bar{u}_i + \bar{v}_j - d_{ij} > 0)$ şi se consideră ciclul corespunzător care va conduce la noua soluţie modificată.

Testarea optimalităţii s-a efectuat în tabelul 2.33.

Căsuţele haşurate corespund variabilelor bazice (pentru care diferenţele $\bar{u}_i + \bar{v}_j - d_{ij} = 0$). Soluţia nu este optimă căci sînt diferenţe strict pozitive. Diferenţei maxime îi corespunde ciclul marcat în tabelul soluţiei iniţiale*.

* Dacă am fi avut cea mai mare diferenţă pozitivă în căsuţa (D_1, S_5) atunci am fi obţinut ciclul $(D_1, S_5) \to (D_2, S_5) \to (D_2, S_1) \to (D_3, S_1) \to (D_3, S_4) \to (D_1, S_4) \to (D_1, S_5)$ ceea ce dovedeşte că nu totdeauna se obţine un ciclu simplu ca cel obţinut de noi anterior.

Scanned with OKEN Scanner

Tabelul 2.33

		$v_1=7$	$v_2=5$	$v_3=0$	$v_4=6$	$v_5=3$	$v_6=0$
	D_i \ S_j	S_1	S_2	S_3	S_4	S_5	S_6
$u_1=0$	D_1	-7	-3	-11		-9	
$u_2=-1$	D_2		-14	-16	-5		-1
$u_3=5$	D_3					2	5

Noua soluţie de bază este dată în tabelul 2.34.

Tabelul 2.34

D_i \ S_j	S_1	S_2	S_3	S_4	S_5	S_6	Disponibil
D_1				120		90	210
D_2	120				80		200
D_3	30	180	50			10	270
Necesar	150	180	50	120	80	100	680

Valoarea funcţiei obiectiv corespunzătoare acestei soluţii este $Z^{(1)} = 4010$ tone-km.

Testăm acum optimalitatea acestei soluţii în tabelul 2.35.

Tabelul 2.35

		$v_1=12$	$v_2=10$	$v_3=5$	$v_4=6$	$v_5=8$	$v_6=0$
	D_i \ S_j	S_1	S_2	S_3	S_4	S_5	S_6
$u_1=0$	D_1	-2	2	-6		-4	
$u_2=-6$	D_2		-14	-16	-10		-6
$u_3=0$	D_3				-5	2	

Există o diferenţă pozitivă în căsuţa (D_3, S_5) căreia îi corespunde ciclul $(D_3, S_5) \to (D_3, S_1) \to (D_2, S_1) \to (D_2, S_5) \to (D_3, S_5)$ şi se obţine noua soluţie de bază (vezi tabelul 2.36)

Scanned with OKEN Scanner

Tabelul 2.36

D_i \ S_j	S_1	S_2	S_3	S_4	S_5	S_6	Disponibil
D_1		*		120		90 →	210
D_2	150	↑			50	↓	200
		180 ←				10	
D_3			50		30		270
Necesar	150	180	50		80	100	680

Valoarea funcţiei obiectiv corespunzătoare este

$$z^{(2)} = 3950 \text{ tone-km.}$$

Testăm optimalitatea acestei soluţii în tabelul 2.37.

Tabelul 2.37

			$v_1=10$	$v_2=10$	$v_3=5$	$v_4=6$	$v_5=6$	$v_6=0$
	D_i \ S_j		S_1	S_2	S_3	S_4	S_5	S_6
$u_1 = 0$	D_1		−4	2	−6		−6	
$u_2 = -4$	D_2			−12	−14	−8		−4
$u_3 = 0$	D_3		−2			−5		

Soluţia nu este optimă deoarece în căsuţa (D_1, S_2) se află o diferenţă strict pozitivă. Pe baza ciclului corespunzător acestei diferenţe pozitive obţinem noua soluţie (vezi tabelul 2.38).

Tabelul 2.38

D_i \ S_j	S_1	S_2	S_3	S_4	S_5	S_6	Disponibil
D_1		90		120			210
D_2	150				50		200
D_3		90	50		30	100	270
Necesar	150	180	50	120	80	100	680

Valoarea funcţiei obiectiv corespunzătoare este:

$$z^{(3)} = 3770 \text{ tone-km.}$$

Testarea optimalităţii ne conduce la:

Tabelul 2.39

			$v_1=8$	$v_2=8$	$v_3=3$	$v_4=6$	$v_5=4$	$v_6=-2$
	D_i \ S_j		S_1	S_2	S_3	S_4	S_5	S_6
$u_1 = 0$	D_1		−6		−8		−8	−2
$u_2 = -2$	D_2			−12	−14	−2		−4
$u_3 = 2$	D_3		−2			−3		

Întrucît toate diferenţele $\bar{u}_i + \bar{v}_j - d_{ij} \leqslant 0$ soluţia obţinută este optimă, deci planul optim de transport este

$x_{12} = 90$ tone
$x_{14} = 120$ tone
$x_{21} = 150$ tone
$x_{25} = 50$ tone
$x_{32} = 90$ tone
$x_{33} = 50$ tone
$x_{35} = 30$ tone

Rezultă de asemenea că se va păstra cantitatea de 100 tone ciment disponibil în depozitul D_3.

2. Să rezolvăm problema în ipoteza că nu mai funcţionează depozitul D_2. Avem atunci o problemă de transport neechilibrată în care necesarul este mai mare decît disponibilul, pe care o echilibrăm introducînd un depozit fictiv cu distanţe foarte mari. La datele problemei este suficient să luăm de exemplu distanţele respective egale cu 100 km.

Obţinem astfel tabelul 2.40.

Tabelul 2.40

D_i \ S_j	S_1	S_2	S_3	S_4	S_5	Disponibil
D_1	14	8	11	6	12	210
$\tilde{D}_2$	100	100	100	100	100	100
D_3	12	10	5	11	6	270
Necesar	150	180	50	120	80	580

Am notat $\tilde{D}_2$ depozitul fictiv în care am plasat cantitatea de 100 tone care echilibrează necesarul cu disponibilul.

Folosind metoda elementului minim din tabel obţinem soluţia iniţială de bază :

Tabelul 2.41

D_i \ S_j	S_1	S_2	S_3	S_4	S_5	Disponibil
D_1		90		120		210
$\tilde{D}_2$	100					100
D_3	50	90	50		80	270
Necesar	150	180	50	120	80	580

Valoarea funcţiei obiectiv corespunzătoare acestei soluţii este :

$$z^0 = 90 \cdot 8 + 120 \cdot 6 + 100 \cdot 100 + 50 \cdot 12 + 90 \cdot 10 + 50 \cdot 5 + 80 \cdot 6 = 4.670 \text{ tone-km.}$$

Testăm optimalitatea soluţiei în tabelul de mai jos :

Tabelul 2.42

		$v_1=10$	$v_2=8$	$v_3=3$	$v_4=6$	$v_5=4$
	D_i \ S_j	S_1	S_2	S_3	S_4	S_5
$u_1 = 0$	D_1	−4		−8		−8
$u_2 = 90$	D_2		−2	−7	−4	−6
$u_3 = 2$	D_3				−3	

Scanned with OKEN Scanner

Toate diferenţele $\bar{u}_i + \bar{v}_j - d_{ij} \leqslant 0$, deci soluţia obţinută este optimă. Prin urmare planul optim de transport este următorul:

$x_{12} = 90$ tone $\qquad x_{32} = 90$ tone

$x_{14} = 120$ tone $\qquad x_{33} = 50$ tone

$x_{31} = 50$ tone $\qquad x_{35} = 80$ tone

Şantierul S_1 primeşte deci numai 50 tone de la depozitul D_3, deficitul de 100 tone ciment fiind repartizat numai la şantierul S_1.

2.11.2. Alte domenii de aplicare a problemelor de programare liniară de tip transport

O gamă largă de fenomene economice pot fi reprezentate matematic printr-un model de programare liniară de tip transport. Problemele ce se înscriu în această clasă se rezolvă fie prin utilizarea directă a modelului matematic expus în paragraful precedent, fie prin adaptarea acestuia în vederea realizării izomorfismului cu fenomenul analizat.

2.11.2.1. *Probleme de repartiţie pe obiective*

În prima categorie se încadrează aşa-numitele probleme de repartiţie pe obiective. Pentru elucidarea modului în care se formulează o astfel de problemă vom prezenta pe scurt două aplicaţii de acest tip.

1. *Repartiţia unor produse sau operaţiuni pe utilaje*

Informaţiile iniţiale pe baza cărora se scrie modelul matematic corespunzător unei astfel de probleme sînt prezentate în tabelul 2.43.

Tabelul 2.43

Utilaje \ Produse	P_1 P_2 ... P_n	Capacit. disponibile (buc.)
U_1	c_{11} c_{12} . . . c_{1n}	a_1
U_2	c_{21} c_{22} . . . c_{2n}	a_2
.	. . .	.
.	. . .	.
.	. . .	.
.	. . .	.
U_m	c_{m1} c_{m2} . . . c_{mn}	a_m
Producţia planificată (buc.)	b_1 b_2 . . . b_n	$\sum_1^m a_i$ / $\sum_1^n b_j$

Pentru elaborarea programului de producţie se pot utiliza diferite criterii de optimizare. Astfel, dacă ne interesează minimizarea costurilor de fabricaţie, atunci coeficienţii c_{ij} reprezintă costul unitar al prelucrării produsului P_j pe utilajul U_i. În acelaşi fel se poate pune problema pentru minimizarea rebuturilor sau a consumurilor de materiale.

În cazul în care coeficientul c_{ij} $(i = 1,2,\ldots,m\,;\ j = 1,2,\ldots,n)$ se defineşte ca fiind productivitatea ce se realizează pe utilajul U_i cînd se fabrică produsul P_j, atunci se pune problema maximizării producţiei totale fabricate în unitatea de timp. Modelul matematic corespunzător acestei probleme a fost expus în paragraful precedent. Prin rezolvarea lui se determină nivelul variabilelor x_{ij} care reprezintă cantitatea din produsul P_j ce trebuie fabricată pe utilajul U_i. Trebuie precizat că modelul are la bază poteza că produsul P_j se poate fabrica independent pe orice utilaj U_i.

2. *Repartiţia planului de fabricaţie al unui produs omogen pe unităţi ale unei centrale*

La elaborarea planului de producţie al unei centrale industriale se poate pune problema minimizării costului de fabricaţie pentru un produs omogen. În acest caz datele din tabelul 2.43 au următoarea semnificaţie :

U_i reprezintă unitatea $i(i = 1,2,\ldots,m)$ a centralei, iar a_i capacitatea de producţie a acesteia ; P_j — sortul $j(j = 1,2,\ldots,n)$ din produsul omogen considerat, iar b_j — cantitatea ce trebuie produsă din sortul j, c_{ij} reprezintă costul unitar de fabricaţie pentru sortul P_j la unitatea U_i.

Repartiţia optimă a planului de producţie constă în determinarea cantităţilor x_{ij} ce urmează să le fabrice unitatea U_i din sortul P_j în condiţiile impuse de modelul matematic al problemei de transport clasice.

2.11.2.2. *Probleme de transport generalizate*

Pentru planificarea unor activităţi economice, realizarea izomorfismului între sistemul real şi modelul ataşat acestuia impune adaptarea modelului de transport. Astfel de situaţii apar în cazul cînd unele restricţii ale problemei sînt inegalităţi, sau cînd indicatorii a_i şi b_j sînt exprimaţi în unităţi de măsură diferite. În consecinţă, coeficienţii restricţiilor sînt diferiţi de unu iar modelul ar forma :

$$\max(\min)[\mathbf{Z}(x)] = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

cu restricţiile :

$$\sum_{j=1}^{n} d_{ij}x_{ij} \leqslant a_i \qquad (i = 1,2,\ldots,m)$$

$$\sum_{i=1}^{m} e_{ij}x_{ij} = b_j \qquad (j = 1,2,\ldots,n)$$

$$x_{ij} \geqslant 0$$

Scanned with OKEN Scanner

Făcînd substituţiile $d_{ij}x_{ij} = y_{ij}$, $e_{ij}/d_{ij} = g_{ij}$ şi $c_{ij}/d_{ij} = f_{ij}$ sau $e_{ij}x_{ij} = y_{ij}$, $d_{ij}/e_{ij} = k_{ij}$ şi $c_{ij}/e_{ij} = h_{ij}$ se ajunge la unul dintre modelele de mai jos :

$$\max(\min)[\mathbf{Z}(y)] = \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}y_{ij} \qquad \max(\min)[\mathbf{Z}(y)] = \sum_{i=1}^{m}\sum_{j=1}^{n} h_{ij}y_{ij}$$

$$\sum_{j=1}^{n} y_{ij} \leqslant a_i \quad (i = 1,2,\ldots,m) \qquad \sum_{j=1}^{n} k_{ij}y_{ij} \leqslant a_i \quad (i = 1,2,\ldots,m)$$

sau

$$\sum_{i=1}^{m} g_{ij}y_{ij} = b_j \quad (j=1,2,\ldots,n) \qquad \sum_{i=1}^{m} y_{ij} = b_j \quad (j=1,2,\ldots,n)$$

$$y_{ij} \geqslant 0 \qquad y_{ij} \geqslant 0$$

Deşi în modelul de mai sus coeficienţii $g_{ij} \neq 1$ sau $k_{ij} \neq 1$ totuşi el are o formă mai convenabilă pentru aplicarea algoritmului de rezolvare a problemelor de programare liniară de tip transport.

Vom prezenta metoda de rezolvare a unui model de acest tip pe baza problemei ce rezultă din datele tabelului 2.44.

Tabelul 2. 44

Produse / Utilaje	P_1	P_2	P_3	P_4	Timp disponibil în ore (a_i)
R_1	4 | 6	7 | 10	8 | 5	5 | 15	500
R_2	6 | 12	3 | 8	7 | 3	4 | 11	700
R_3	5 | 10	6 | 9	5 | 6	3 | 7	1100
Producţia planificată în bucăţi (b_j)	2 500	4 000	3 800	5 000	

În afară de timpul disponibil al utilajelor şi producţia planificată, tabelul conţine randamentul orar W_{ij} ce se realizează pe utilajul R_i la fabricarea produsului P_j, precum şi beneficiul c_{ij} ce se obţine la livrarea unei unităţi din produsul P_j fabricat pe utilajul R_i (coeficienţii c_{ij} sînt trecuţi deasupra diagonalei principale a fiecărei căsuţe).

Scanned with OKEN Scanner

Modelul matematic corespunzător acestei probleme are forma :

$$\max\ [\mathbf{Z}(x)] = \sum_{i=1}^{3} \sum_{j=1}^{4} c_{ij} x_{ij}$$

cu restricţiile

$$\sum_{j=1}^{4} \frac{1}{W_{ij}}\, x_{ij} \leqslant a_i \quad (i = 1,2,3) \tag{2.30}$$

$$\sum_{i=1}^{3} x_{ij} = b_j \qquad (j=1,2,3,4)$$

$$x_{ij} \geqslant 0$$

unde x_{ij} reprezintă numărul de piese de tipul P_j ce se fabrică pe utilajul R_i.

Ca şi la problemele clasice de transport, algoritmul de aflare a soluţiei optime parcurge cele două etape cunoscute din paragraful precedent cărora li se aplică nişte transformări impuse de model.

Astfel, la determinarea soluţiei iniţiale, pentru stabilirea nivelului variabilei x_{lk} ce intră în bază se foloseşte relaţia :

$$x_{lk} = \min\ (a_l \,.\, W_{lk},\ b_k)\ (l = 1,2,3)$$
$$(k = 1,2,3,4).$$

Dacă $x_{lk} = a_l \,.\, W_{lk}$, atunci $x_{lj} = 0$ pentru $j \neq k$ iar în cazul în care $x_{lk} = b_k$, atunci $x_{ik} = 0$ pentru $i \neq l$. Utilizînd această relaţie în cadrul metodei elementului maxim din tabel se obţine următoarea soluţie iniţială :

Tabelul 2.45

R_i \ P_j	P_1	P_2	P_3	P_4	x_{i5}
R_1			2 500		0
R_2	2 500		1 300	641,66	0
R_3		4 000		4358,34	33

Tabelul 2.45 conţine o coloană suplimentară pentru a pune în evidenţă participarea variabilelor de ecart x_{i5} $(i = 1, 2, 3)$, care se ataşează primelor restricţii, la determinarea soluţiilor de bază. Deoarece componentele acestei coloane reprezintă timpul nefolosit la fiecare utilaj atunci $W_{i5} = 1$, şi $c_{i5} = 0$.

Scanned with OKEN Scanner

Pentru optimizarea soluției inițiale se aplică metoda distributiv-modificată. Plecînd de la duala formei standard a problemei (2.30) rezultă sistemul de ecuații:

$$\frac{1}{W_{ij}} u_i + v_j = c_{ij} \quad (i, j \in J_B) \tag{2.31}$$

Pe baza acestuia se stabilește criteriul de optimalitate.

$$\Delta_{ij} = c_{ij} - \left(\frac{1}{W_{ij}} \bar{u}_i + \bar{v}_j \right) \leqslant 0 \quad (i = 1,2,3),$$
$$(j = 1,2, \ldots,5)$$

unde $\bar{u}_i$ și $\bar{v}_j$ se obțin prin rezolvarea sistemului (2.31).

Aplicînd relația (2.31) soluției inițiale dată în tabelul 2.45 se obține următorul sistem de ecuații:

Tabelul 2.46

$\bar{u}_i$ \ $\bar{v}_j$	$\frac{61}{12}$	6	$\frac{10}{3}$	3	0
70/3	−179/6	−40/3	0	20/3	−70/3
11	0	−35	0	0	−11
0	−5/6	0	10	0	0

$$u_1 + 5v_3 = 40 \qquad u_3 + 9v_2 = 54$$
$$u_2 + 12v_1 = 72 \qquad u_3 + 7v_4 = 21$$
$$u_2 + 3v_3 = 21 \qquad u_3 + v_5 = 0$$
$$u_2 + 11v_4 = 44$$

Întrucît timpul disponibil al utilajului R_3 nu a fost utilizat în întregime ($x_{35} = 33$), în conformitate cu teorema ecarturilor complementare $u_3 = 0$ (deci și $v_5 = 0$). Soluția sistemului de ecuații de mai sus și diferențele Δ_{ij} sînt date în tabelul 2.46. Deoarece $\Delta_{33} = 10 > 0$, variabila x_{33} va intra în noua soluție. Pentru a stabili nivelul variabilei x_{33} se formează un ciclu după principiile expuse în paragraful precedent.

	P_3	P_4	
R_2	$1300 - \alpha$ ←	$641{,}67 + \frac{11}{3}\alpha$	$\begin{cases} 1300 - \alpha \geqslant 0 & \alpha \leqslant 1300 \\ 4358{,}3 - \frac{11}{3}\alpha \geqslant 0; & \alpha \leqslant 1188{,}6 \end{cases}$
R_3	α ←	$4358{,}3 - \frac{11}{3}\alpha$	$\alpha = \min\ (1300;\ 1188{,}6) = 1188{,}6$

Scanned with OKEN Scanner

Nivelul variabilei $x_{33} = \alpha$ se determină ţinînd seama de cele două condiţii de nenegativitate scrise mai sus, precum şi de coeficientul $\theta_{24} = W_{24}/W_{23} = 11/3$ prin care se asigură, la resursa R_2, transferul aceluiaşi volum de timp de la produsul P_3 la P_4.

Noua soluţie este dată în tabelul 2.47 iar diferenţele Δ_{ij} corespunzătoare acesteia în tabelul 2.48.

Tabelul 2.47

0	0	2500	0	0
2 500	0	111	5 000	0
0	4 000	1 189	0	4 572

Tabelul 2.48

u_i \ v_1	11/12	6	5	38/11	0
15	−24	− 5	0	85/11	−15
6	0	−30	0	0	− 6
0	− 5	0	0	−35/11	0

Se constată că $\Delta_{14} = 85/11$, deci trebuie introdusă în bază variabila $x_{14} = \alpha$, unde α se determină analizînd relaţiile din cadrul ciclului :

	P_3		P_4	
R_1	$2500 - \frac{5}{15}\alpha$	←	α	$2500 - \frac{5}{15}\alpha \geqslant 0$; $\alpha \leqslant 7500$
	↓		↑	$5000 - \alpha \geqslant 0$; $\alpha \leqslant 5\,000$
R_2	$111 + \frac{5}{15}\alpha$	→	$5000 - \alpha$	$\alpha = \min\ (5000;\ 7500) = 5000$

Se observă că în cadrul acestei iteraţii, trebuie să se ţină seama de faptul că $x_{25} = 0$ şi de aceea nu putem repartiza resursei R_2 cantitatea $x_{23} = 111 + 5000/3 = 1777{,}67$ $(1777{,}67/3 > 111/3 + 5000/11)$.

Volumul maxim de timp disponibil pe utilajul R_2 pentru fabricarea produselor P_3 şi P_4 este $111/3 + 5\,000/11 = 491{,}54$, deci $x_{23} = 491{,}54 \times 3 = 1475$, iar diferenţa $\Delta x_{23} = 1\,777{,}67 - 1\,475 = 303$, se va aloca utilajului R_3 unde există timp disponibil, deci $x_{33} = 1189 + 303 = 1\,492$. Noua soluţie dată în tabelul 2.49 este optimă deoarece toate diferenţele $\Delta_{ij} \leqslant 0$ (vezi tabelul 2.50).

Tabelul 2.49

0	0	833	5 000	0
2 500	0	1 475	0	0
	4 000	1 492	0	407

Tabelul 2.50

u_i \ v_1	11/2	6	5	4	0
15	−24	− 5	0	0	−15
6	0	−30	0	− 6	− 6
0	− 5	0	0	− 7	0

Trebuie precizat că în fiecare iteraţie, indicatorii u_i reprezintă preţurile umbră ataşate resurselor R_i $(i = 1, 2, 3)$ iar v_j — preţurile umbră ale produselor P_j $(j = 1, 2, 3, 4)$.

Scanned with OKEN Scanner

Capitolul III

TEORIA GRAFELOR

3.1. Obiectul teoriei grafelor

Teoria grafelor s-a constituit ca un capitol de-sine-stătător în cercetarea operațională datorită multiplelor avantaje pe care le oferă activității de planificare și analiză economică. Modelele matematice din această clasă sînt de regulă liniare, dar spre deosebire de modelele de programare liniară, spre exemplu, acestea au proprietăți structurale care permit reprezentarea mai completă a fenomenelor analizate și totodată creșterea gradului de eficacitate a procesului de optimizare. În elaborarea programelor de producție, investiții, transport, desfacere etc. la unități economice cu activități deosebit de complexe utilizarea grafelor și a modelelor matematice corespunzătoare acestora constituie o necesitate practică.

În aplicarea teoriei grafelor se utilizează conceptul de graf care constituie o formă specifică de reprezentare a fenomenelor.

Cu ajutorul grafului se surprind și se analizează în detaliu legăturile dintre componentele sistemului economic considerat, se elaborează cu mai multă ușurință și precizie modelul matematic corespunzător.

Pentru a ușura înțelegerea conceptelor de bază ale teoriei grafelor, a semnificației economice a elementelor constitutive ale unui graf și a modelului matematic atașat acestuia vom prezenta cîteva aplicații.

Exemplul 3.1. În acest exemplu vom analiza într-o altă perspectivă problema generală de transport prezentată în paragraful 2.2. Considerăm că în această problemă coeficientul c_{ij} reprezintă capacitatea maximă de transport pe ruta (i, j). Prin urmare, dacă se face abstracție de costurile de transport, atunci se poate formula o problemă de programare liniară prin care urmează să se determine cantitatea maximă ce trebuie transportată pe diverse rute pentru satisfacerea cererii de consum. Modelul matematic corespunzător acestei probleme are forma :

$$\max \ [Z(x_{ij})] = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}$$

Scanned with OKEN Scanner

în condițiile

$$\sum_{j=1}^{n} x_{ij} \leqslant a_i \quad (i = 1,2, \ldots,m)$$

$$\sum_{i=1}^{m} x_{ij} \geqslant b_j \; (j = 1,2,\ldots,n)$$

$$x_{ij} \leqslant c_{ij} (i = 1,2,\ldots,m \; ; \; j = 1,2,\ldots,n)$$

$$x_{ij} \geqslant 0.$$

Soluția optimă a unei astfel de probleme se poate obține mai ușor cu ajutorul teoriei grafelor. În acest scop se construiește un graf denumit *rețea de transport* folosind următoarele reguli : se reprezintă în plan prin *puncte* centrele producătoare $\mathbf{A}_1,\ldots,\mathbf{A}_m$ și centrele consumatoare $\mathbf{B}_1\ldots,\mathbf{B}_n$ care se unesc prin segmente orientate $(\mathbf{A}_i, \mathbf{B}_j)$ denumite *arce*, cărora le atașăm *capacitatea* c_{ij}. În cazul în care un centru producător $\mathbf{A}_i$ nu are mijloace de transport disponibile către centrul $\mathbf{B}_j$ capacitatea arcului corespunzător $(\mathbf{A}_i, \mathbf{B}_j)$ se va lua egală cu zero. Adăugăm grafului format o „intrare" **A** legată prin arce de fiecare dintre punctele $\mathbf{A}_1\ldots\mathbf{A}_m$ și numai de acestea și de asemenea o „ieșire" **B** în care sosește cîte un arc de la fiecare din punctele $\mathbf{B}_1,\ldots,\mathbf{B}_n$, și numai din acestea. Capacitatea unui arc de forma $(\mathbf{A},\mathbf{A}_i,)$ se va lua egală cu a_i, în timp ce capacitatea unui arc de forma $(\mathbf{B}_j, \mathbf{B})$ se va considera egală cu b_j. În acest fel s-a obținut rețeaua de transport din figura 3.1.

Pe baza unui algoritm specific, care va fi prezentată în paragraful (3.5) se determină fluxul maxim în rețea care are semnificația cantității maxime de produs care pleacă din punctul **A** și sosește în punctul **B**.

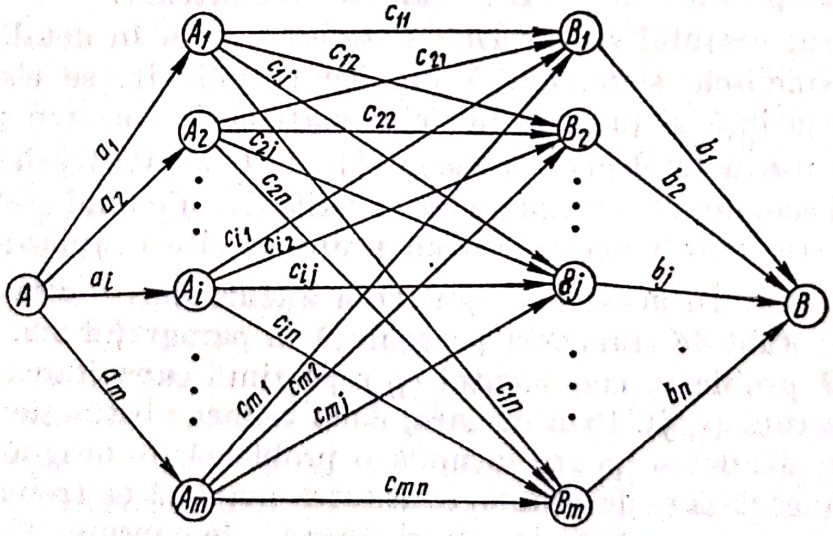


Fig. 3.1

Folosind aceeași tehnică de modelare se poate rezolva așa-numita *problemă de transport cu centre intermediare* în care între producătorii $\mathbf{A}_i$ și consumatorii $\mathbf{B}_i$ se interpun centrele intermediare (depozite) $\mathbf{D}_i$.

Scanned with OKEN Scanner

Exemplul 3.2. În cadrul unei întreprinderi industriale se pune problema stabilirii variantei tehnologice optime de prelucrare a unui produs. La prelucrarea produsului participă, într-o ordine prestabilită, mai multe activităţi. Dacă reprezentăm grafic activităţile a_j prin segmente orientate în sensul de prelucrare a produsului, denumite arce, atunci fiecare variantă tehnologică va fi constituită dintr-o succesiune de arce (vezi figura 3.2).

Graful din figura 3.2. conţine mai multe drumuri care unesc vîrful x_1 cu x_6, fiecare dintre acestea constituind o variantă tehnologică prin care poate fi fabricat produsul respectiv.

Indicatorii $v(x_i, x_j)$ ataşaţi activităţii reprezentate pe arcul (x_i, x_j) au semnificaţia costurilor parţiale de prelucrare a produsului.

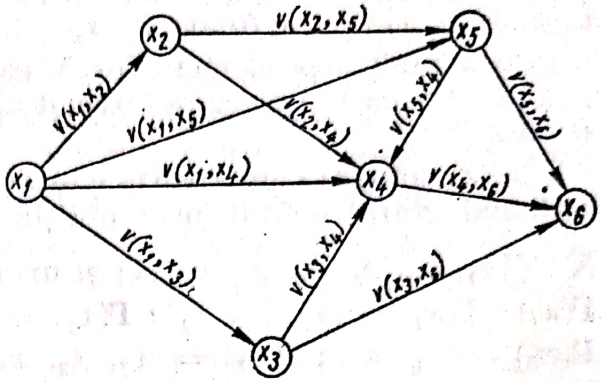


Fig. 3.2.

Prin urmare se pune problema determinării variantei tehnologice căreia îi corespunde un cost total de prelucrare minim. O astfel de problemă se rezolvă cu ajutorul unui algoritm care permite aflarea drumului de valoare minimă.

Exemplul 3.3. Pentru determinarea timpului total de execuţie a unui proiect complex (construirea unui obiectiv economic, repararea utilajelor etc.) acesta se reprezintă printr-un graf care pune în evidenţă activităţile şi legăturile tehnologice şi economice dintre activităţi. Fiecare arc al grafului constituie o activitate a proiectului, iar indicatorii $v(x_i, x_j)$ ataşaţi acestora reprezintă duratele de execuţie a activităţilor (x_i, x_j). Durata totală de execuţie a proiectului este dată de drumul de valoare maximă care uneşte vîrful iniţial şi final al grafului. Algoritmul de rezolvare a acestei probleme va fi prezentat în paragraful (3.4.2).

Sfera de aplicare a teoriei grafelor este mult mai largă, de aceea unele aspecte legate de utilizarea lor practică vor fi tratate în cadrul acestui capitol.

3.2. Concepte fundamentale

Pe baza interpretării exemplelor prezentate în paragraful precedent putem trage concluzia că un graf este definit prin mulţimea vîrfurilor $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ şi mulţimea legăturilor dintre acestea. Dacă notăm cu Γ *aplicaţia multivocă a mulţimii* $\mathbf{X}$ *în ea însăşi*, care pune în evidenţă tocmai conexiunile dintre vîrfurile x_i ale grafului, atunci *graful* $\mathbf{G}$ *se defineşte prin cuplul* $(\mathbf{X}, \Gamma)$, deci $\mathbf{G} = (\mathbf{X}, \Gamma)$.

Scanned with OKEN Scanner

Pentru mai multă claritate, grafului **G** i se ataşează un grafic construit astfel : fiecare element $x_i \in \mathbf{X}$ se reprezintă printr-un punct denumit vîrf (nod), iar conexiunea dintre două vîrfuri $x_i \in \mathbf{X}$ şi $x_j \in \mathbf{X}$ se reprezintă printr-un segment orientat de la x_i la x_j dacă $x_j \in \Gamma(x_i)$. Perechea ordonată (x_i, x_j) se numeşte *arc al grafului G* şi are pe x_i ca extremitate iniţială, iar pe x_j cu extremitate finală. Prin urmare *aplicaţia* $\Gamma(x_i)$ pune în evidenţă mulţimea punctelor x_j legate cu x_i prin arce care au extremitatea iniţială în x_i şi cea finală în x_j.

Dacă mulţimea elementelor **X** este finită, atunci graful **G** este finit. Am făcut această precizare întrucît aplicaţiile la economie utilizează grafe finite.

În continuare vom analiza conceptele fundamentale ale teoriei grafelor utilizînd graful definit prin mulţimea

$\mathbf{X} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7\}$ şi următoarele aplicaţii
$\Gamma(x_i)$: $\Gamma(x_1) = \{x_2, x_3, x_4\}$; $\Gamma(x_2) = \{x_1, x_2, x_3, x_5\}$;
$\Gamma(x_3) = \{x_6, x_7\}$; $\Gamma(x_4) = \{x_2, x_3, x_5\}$; $\Gamma(x_5) = \{x_6\}$;
$\Gamma(x_6) = \{x_3, x_4, x_7\}$ şi $\Gamma(x_7) = \Phi$. Acestui graf îi corespunde reprezentarea dată în figura 3.3.

Graful din figura 3.3. s-a obţinut unind punctele ce formează mulţimea **X** prin arce definite de aplicaţiile $\Gamma(x_i)$. Astfel, aplicaţia $\Gamma(x_3)$ arată că vîrfurile x_6 şi x_7 sînt legate cu x_3 prin arce care au extremitatea iniţială în x_3. Se poate utiliza şi conceptul de *aplicaţie inversă în vîrful* x_j, care se notează cu $\Gamma^{-1}(x_j)$ şi se defineşte astfel

$$\Gamma^{-1}(x_j) = \{x_i : x_j \in \Gamma(x_i)\}$$

Aplicaţia $\Gamma^{-1}(x_j)$ pune în evidenţă mulţimea vîrfurilor x_i legate cu x_j prin arce care au extremitatea finală în x_j. Prin urmare $\Gamma^{-1}(x_3) = \{x_1, x_2, x_4, x_6\}$.

Se constată că un graf $\mathbf{G} = (\mathbf{X}, \Gamma)$ poate fi definit prin cuplul **X** şi **U**, adică $\mathbf{G} = (\mathbf{X}, \mathbf{U})$.

Cu **U** s-a notat mulţimea arcelor $u_{ij} = (x_i, x_j)$ a grafului **G**, adică :

$$\mathbf{U} = \{(x_i, x_j)/x_i, x_j \in \mathbf{X}, x_j \in \Gamma(x_i)\}$$

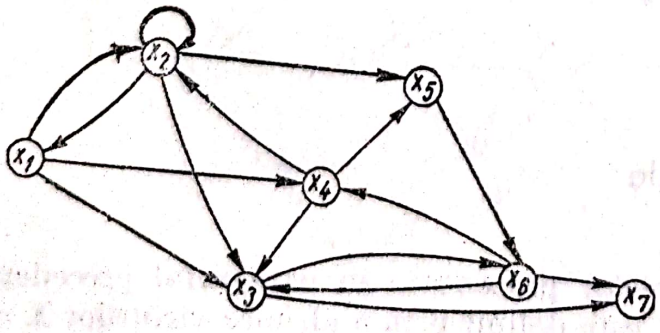


Fig. 3.3.

Două arce ale grafului **G** se numesc adiacente dacă au o extremitate comună, iar vîrfurile x_i şi x_j sînt adiacente, dacă există un arc care le uneşte.

Fie $\mathbf{A} \subset \mathbf{X}$ o submulţime de vîrfuri ale grafului **G**. Un arc $u_j \in \mathbf{U}$ se numeşte *incident spre exterior mulţimii* **A**, dacă extremitatea iniţială a lui u_{ij} este în **A**, iar extremitatea finală nu, deci $x_i \in \mathbf{A}$ iar $x_j \notin \mathbf{A}$. Arcul u_{ij} este *incident spre interior mulţimii* **A** dacă extremitatea

finală a lui u_{ij} este în $\mathbf{A}$, iar cea inițială nu, adică atunci cînd $x_i \notin \mathbf{A}$ și $x_j \in \mathbf{A}$.

În particular, un arc oarecare u este incident spre exterior (interior) unui vîrf x, dacă este incident spre exterior (interior) mulțimii $\{x\}$; aceasta revine la a spune că extremitatea inițială (finală) a arcului u, este x, iar extremitatea finală (inițială) este diferită de x.

Arcul care are ambele extremități într-un vîrf x_i se numește buclă.

Mulțimea arcelor incidente spre exterior submulțimii $\mathbf{A}$ (vîrfului x), se notează cu $\mathbf{U}_{\mathbf{A}}^{+}$ (respectiv $\mathbf{U}_{x}^{+}$) iar mulțimea arcelor incidente spre interior mulțimii $\mathbf{A}$ (vîrfului x) se notează cu $\mathbf{U}_{\mathbf{A}}^{-}$ (respectiv $\mathbf{U}_{x}^{-}$).

Astfel în graful din fig. 3.3. arcele (x_1, x_4) și (x_4, x_5) sînt adiacente, în timp ce arcele (x_1, x_2), (x_3, x_6) nu sînt, vîrfurile x_4 și x_6 sînt adiacente, dar vîrfurile x_1 și x_6 nu sînt.

Fie $\mathbf{A} = \{x_2, x_3, x_6, x_5\}$. Arcul (x_3, x_6), spre exemplu, este incident spre exterior mulțimii $\mathbf{A}$, întrucît $x_3 \in \mathbf{A}$ iar $x_6 \notin \mathbf{A}$. Totalitatea arcelor incidente spre exterior acestei mulțimi formează mulțimea $\mathbf{U}_{\mathbf{A}}^{+}$:

$$\mathbf{U}_{\mathbf{A}}^{+} = \{(x_2, x_1), (x_4, x_2), (x_5, x_6), (x_3, x_6), (x_3, x_7)\}$$

Arcul (x_1, x_4), spre exemplu, este incident spre interior mulțimii $\mathbf{A}$ deoarece $x_1 \notin \mathbf{A}$ iar $x_4 \in \mathbf{A}$. Mulțimea $\mathbf{U}_{\mathbf{A}}^{-}$ este constituită din arcele:

$$\mathbf{U}_{\mathbf{A}}^{-} = \{(x_1, x_2), (x_1, x_3), (x_1, x_4), (x_6, x_4), (x_6, x_3)\}.$$

Dacă considerăm că $\mathbf{A} = \{x_2\}$ atunci arcul (x_2, x_3) este incident spre exterior vîrfului x_2, iar arcul (x_4, x_2) este incident spre interior acestuia. Bucla de extremitate finală și inițială x_2 nu este incidentă nici spre exterior nici spre interior lui x_2. Deci:

$$\mathbf{U}_{x_2}^{+} = \{(x_2, x_1), (x_2, x_3), (x_2, x_5)\}; \quad \mathbf{U}_{x_2}^{-} = \{(x_1, x_2), (x_4, x_2)\}$$

În multe raționamente din teoria grafelor este frecventă operația de „ștergere" (eliminare) a unor arce dintr-un graf dat, sau a unor vîrfuri și a tuturor arcelor incidente spre exterior sau interior acestora. Ca urmare a acestor operațiuni se obține un *graf parțial* sau un *subgraf al grafului* $\mathbf{G}$.

Se numește graf parțial al lui $\mathbf{G}$, un graf de forma $\mathbf{G}' = (\mathbf{X}, \mathbf{\Gamma}')$, în care aplicația multivocă $\mathbf{\Gamma}'$ satisface condiția $\mathbf{\Gamma}'(x_i) \subset \mathbf{\Gamma}(x_i)$ dar $\mathbf{\Gamma}'(x_i) \leqslant \mathbf{\Gamma}(x_i)$ pentru orice $x_i \in \mathbf{X}$. Reținem deci că un graf parțial al lui $\mathbf{G}$ are aceleași vîrfuri ca și $\mathbf{G}$, dar mai puține arce.

Se numește subgraf a lui $\mathbf{G}$, un graf de forma $\overline{\mathbf{G}} = (\overline{\mathbf{X}}, \overline{\mathbf{\Gamma}})$ în care $\overline{\mathbf{X}} \subset \mathbf{X}$ iar pentru orice $x_i \in \overline{\mathbf{X}}$ este satisfăcută relația $\overline{\mathbf{\Gamma}}(x_i) = \mathbf{\Gamma}(x_i) \cap \mathbf{X}$. Deci, un subgraf al grafului $\mathbf{G}$ se obține prin eliminarea din $\mathbf{G}$ a unui număr de vîrfuri și a arcelor incidente spre exterior și interior acestora.

Combinînd cele două concepte de mai sus va rezulta un subgraf parțial al unui graf.

Un *graf* $\mathbf{G}$ *se numește complet* dacă orice cuplu de vîrfuri este legat cu cel puțin un arc indiferent de direcția acestuia; dacă orice cuplu (x_i, x_j) este legat prin arce în ambele sensuri adică $(x_i, x_j) \in \mathbf{U}$ și $(x_j, x_i) \in \mathbf{U}$, atunci graful se numește simetric.

Scanned with OKEN Scanner

Oricărui graf **G** i se poate asocia o *matrice booleană* $\mathbf{M_G} = (m_{ij})$ unde :

$$m_{ij} = \begin{cases} 1 \text{ dacă există arcul cu extremitate inițială în } x_i \text{ și cea finală în } x_j ; \\ 0 \text{ dacă vîrfurile } x_i \text{ și } x_j \text{ nu sînt adiacente.} \end{cases}$$

Matricea asociată grafului din figura 3.3. are următoarea structură :

$$\mathbf{M_G} = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Trebuie precizat că un graf este complet determinat prin matricea booleană asociată. Diferitele operații executate asupra unui graf se concretizează în operații corespunzătoare efectuate în matricea $\mathbf{M_G}$. Astfel, matricea asociată unui graf parțial se deduce din matricea grafului inițial prin înlocuirea unor elemente $m_{ij} = 1$ corespunzătoare arcelor care se elimină prin elemente $m_{ij} = 0$. Matricea asociată unui subgraf se deduce din matricea inițială prin eliminarea liniilor și coloanelor corespunzătoare vîrfurilor eliminate.

În aplicațiile teoriei grafelor la economie cel mai frecvent se utilizează conceptul de drum.

Un drum μ *într-un graf* $\mathbf{G} = (\mathbf{X}, \mathbf{U})$ este o succesiune de arce $(u_1, u_2, \ldots, u_p)$ cu proprietatea că extremitatea finală a arcului u_i coincide cu extremitatea inițială a arcului următor.

Extremitatea inițială a drumului μ este prin definiție extremitatea inițială a arcului u_1, în timp ce extremitatea finală a lui μ este extremitatea finală a arcului u_p. Drumul μ se numește simplu, dacă arcele sale sînt distincte ; în caz contrar, el se zice compus. Drumul μ se numește elementar, dacă nu trece de două ori prin același vîrf, în caz contrar spunem că μ este neelementar.

Lungimea $l(\mu)$ a drumului μ este, prin definiție, numărul arcelor componente, adică $l(\mu) = p$.

Un circuit este un drum a cărui extremitate inițială coincide cu cea finală. Un circuit sau un drum care trece o dată și numai o dată prin toate vîrfurile grafului **G** se numește circuit hamiltonian. Un *circuit hamiltonian* are lungimea n, iar un drum hamiltonian are lungimea $n-1$.

Un graf este *tare conex* dacă pentru orice cuplu de vîrfuri x_i și $x_j (x_i \neq x_j)$ există un drum de la x_i spre x_j.

Pe baza grafului din figura 3.3. vom concretiza conceptele definite mai sus :

$\mu_1 = \{(x_1, x_2), (x_2, x_5), (x_5, x_6), (x_6, x_7)\}$ este drum simplu și elementar ;

$\mu_2 = \{(x_1, x_4), (x_4, x_2), (x_2, x_1), (x_1, x_4), (x_4, x_5)\}$ este drum compus ;

Scanned with OKEN Scanner

$\mu_3 = \{(x_3, x_6), (x_6, x_3), (x_3, x_7)\}$ este neelementar;

$\mu_4 = \{(x_1, x_4), (x_4, x_2), (x_2, x_5), (x_5, x_6), (x_6, x_3), (x_3, x_7)\}$ este un drum hamiltonian

3.3. Concepte neorientate

Grafele orientate constituie un mijloc simplu și eficient de reprezentare a multor fenomene concrete de natură economică sau tehnică.

În exemplele analizate este prezentă ideea intuitivă de orientare tradusă prin condiționări, sau prin subordonări.

Dacă examinăm harta căilor ferate ale țării, constatăm că ea se aseamănă foarte mult cu reprezentarea grafică a unui graf; distingem vîrfuri — centrele feroviare importante ale țării, și arce, numai că acestora le lipsește orientarea, chiar și o linie ferată îngustă, putînd fi parcursă în ambele sensuri.

Acest exemplu ne conduce la noțiunea de *graf neorientat*. Un graf neorientat este un cuplu $\mathbf{G} = (\mathbf{X}, \mathbf{M})$, format dintr-o mulțime $\mathbf{X}$ și o mulțime $\mathbf{M}$ de submulțimi nevide ale lui $\mathbf{X}$.

Elementele mulțimii $\mathbf{X}$ se numesc vîrfurile grafului neorientat $\mathbf{G}$, iar elementele mulțimii $\mathbf{M}$ poartă numele de *muchii*.

Dacă $m = \{x, y\}$ este o muchie, vîrfurile x, y se numesc extremitățile muchiei m. O muchie de forma $\{x\}$ se numește bucla de extremitate x. Pentru omogenizarea limbajului, convenim sau spunem că bucla $\{x\}$ are două extremități confundate în vîrful x, și vom nota $\{x, x\}$.

Graful neorientat $\mathbf{G} = (\mathbf{X}, \mathbf{M})$ se numește finit, dacă mulțimea $\mathbf{X}$ este finită. Putem vizualiza un graf neorientat $\mathbf{G}$, dispunînd arbitrar în plan vîrfurile sale și unind printr-o linie continuă acele vîrfuri între care există muchii.

De exemplu, între localitățile x_1, x_2, x_3, x_4, x_5 există următoarele drumuri rutiere:

— o autostradă care trece prin localitățile x_1, x_3, x_4

— o șosea asfaltată ce trece prin x_2, x_4, x_5

— drumuri de țară între x_1 și x_2; între x_1 și x_4; între x_2 și x_5.

Construim un graf neorientat $\mathbf{G} = (\mathbf{X}, \mathbf{M})$ în care:

$$\mathbf{X} = \{x_1, x_2, x_3, x_4, x_5\}$$

$$\mathbf{M} = \{(x_1, x_3), (x_3, x_4), (x_2, x_4), (x_4, x_5), (x_1, x_2), (x_1, x_4), (x_2, x_5)\}$$

În figura 3.4. este dată reprezentarea grafică a acestui graf.

Dintr-un graf orientat $\mathbf{G} = (\mathbf{X}, \Gamma)$ putem obține un graf neorientat $\mathbf{G}$, în felul următor:

— graful neorientat are aceleași vîrfuri ca și cel orientat;

— în graful neorientat există muchia $\{x, y\}$ dacă și numai dacă în graful orientat există arcul (x, y), arcul (y, x) sau amîndouă.

Într-un graf neorientat putem vorbi de vîrfuri adiacente, muchii adiacente ca și într-un graf orientat. Succesiunea de muchii:

Scanned with OKEN Scanner

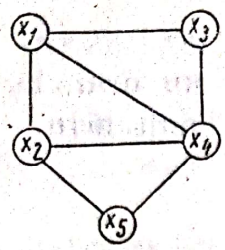


Fig. 3.4.

$1 = (m_1, m_2 \ldots, m_p)$, $p \geqslant 3$ ale grafului neorientat $\mathbf{G} = (\mathbf{X}, \mathbf{M})$ se numește *lanț* dacă muchia m_i, $2 \leqslant i \leqslant p-1$ are o extremitate comună cu muchia m_{i-1} iar cealaltă extremitate comună cu muchia m_{i+1}. Este natural să completăm această definiție spunînd că două muchii formează un lanț, dacă ele sînt adiacente.

Extremitățile lanțului $(m_1, \ldots, m_p)$ sînt extremitățile libere ale muchiilor extreme m_1 și m_p.

Un lanț poate fi dat fie prin muchiile componente, fie prin vîrfurile prin care trece (acestea unesc vîrfurile lanțului). Un lanț ale cărui muchii componente sînt diferite între ele se numește simplu, în caz contrar, compus. Un lanț care trece o singură dată prin vîrfurile sale se numește elementar; în caz contrar, neelementar.

Un lanț ale cărui extremități coincid se numește ciclu. Lungimea unui lanț este numărul muchiilor componente.

3.4. Determinarea drumului de valoare optimă

Una dintre aplicațiile teoriei grafelor constă în determinarea drumului de valoare minimă sau maximă dintre două vîrfuri $x_i \in \mathbf{X}$ și $x_j \in \mathbf{X}$ ale grafului $\mathbf{G} = (\mathbf{X}, \Gamma) = (\mathbf{X}, \mathbf{U})$. În acest caz fiecărui arc $u_i \in \mathbf{U}$ i se asociază un număr real, notat cu $v(u_i) \geqslant 0$ sau $v(x_i, x_j) \geqslant 0$, numit valoarea arcului și care, în funcție de problema economică ce urmează a fi rezolvată, poate avea una dintre următoarele semnificații:

— costul sau durata transportului pe ruta (x_i, x_j);

— costul de fabricație al unui produs pe locul de muncă reprezentat prin arcul (x_i, x_j);

— timpul de muncă necesar executării unui produs (sau productivitatea muncii) la locul de muncă reprezentat prin arcul (x_i, x_j) etc.

Dacă considerăm drumul $\mu = (u_1, u_2, \ldots, u_k)$, atunci mărimea $v(\mu)$ definită prin relația:

$$v(\mu) = \sum_{i=1}^{k} v(u_i)$$

se numește valoarea drumului μ. De regulă interesează valoarea optimă a drumului format din succesiunea de arce $u_i \in \mathbf{U}$ care leagă nodul inițial x_1 cu nodul final x_n. În continuare vom prezenta două dintre metodele de aflare a drumului de valoare optimă într-un graf $\mathbf{G} = (\mathbf{X}, \Gamma)$ fără circuite.

3.4.1. Algoritmul Ford

Pentru aflarea drumului de valoare minimă între două vîrfuri x_1 și x_n, utilizarea acestui algoritm presupune parcurgerea următoarelor etape:

1. Fiecărui vîrf x_i al grafului $\mathbf{G}$ i se atașează o variabilă λ_i. Se încep cu $\lambda_1 = 0$ și $\lambda_i = +\infty$ pentru $i = 2, \ldots, n$. Variabila λ_i are semnificația de valoare a drumului de la vîrful x_1 pînă la vîrful x_n.

Scanned with OKEN Scanner

2. Se caută un arc (x_i, x_j), pentru care $\lambda_j - \lambda_i > v(x_i, x_j)$. Dacă un astfel de arc a fost găsit, se înlocuieşte λ_j cu $\lambda_j' = \lambda_i + v(x_i, x_j) < \lambda_j$. (3.1)

Dacă există mai multe variante pentru relaţia (3.1) care permit micşorarea variabilei λ_j, atunci se alege λ_j' cu valoarea cea mai mică. Se continuă în acest mod pînă cînd nici un arc nu mai permite micşorarea valorilor variabilelor λ_j.

3. Deoarece pentru fiecare vîrf x_i, valorile luate de variabila λ_i, formează un şir descrescător de numere nenegative, va exista un vîrf x_{p_1} cu proprietatea :

$$\lambda_n - \lambda_{p_1} = v(x_{p_1}, x_n)$$

x_{p_1} fiind vîrful care a contribuit ultima oară la micşorarea valorii variabilei λ_n. Analog, va exista x_{p_2} astfel încît :

$$\lambda_{p_1} - \lambda_{p_2} = v(x_{p_2}, x_{p_1}) \text{ etc.}$$

Numărul vîrfurilor fiind finit, vom obţine în final

$$x_{p_{k+1}} = X_1 \text{ şi } \lambda_{p_k} - \lambda_1 = v(x_1, x_{p_k})$$

Se poate arăta că valoarea minimă a drumului care uneşte vîrful x_1 şi x_n este tocmai λ_n iar drumul μ_0 pentru care $v(\mu_0) = \lambda_n$ este $\mu_0 = (x_1, x_{p_k}, \ldots, x_{p_2}, x_{p_1}, x_n)$. Pentru a demonstra acest lucru se consideră un drum oarecare $\mu = [x_1, x_{k_1}, \ldots, x_{k_{s+1}} = x_n]$ pentru care se calculează :

$$\lambda_{k_1} - 0 \leqslant v(x_1, x_{k_1})$$

$$\lambda_{k_2} - \lambda_{k_1} \leqslant v(x_{k_1}, x_{k_2})$$

$$\lambda_n - \lambda_{k_s} \leqslant v(x_{k_s}, x_n)$$

Însumînd separat termenii acestor relaţii rezultă că $\lambda_n \leqslant v(\mu)$ ceea ce demonstrează că μ_0 este drumul cu valoarea cea mai mică.

De asemenea se poate arăta că algoritmul Ford este echivalent cu rezolvarea sistemului :

$$\begin{cases} \lambda_1 = 0 \\ \lambda_j = \min\limits_{(x_i, x_j) \in U_{\bar{x}j}} [\lambda_i + v(x_i, x_j)] \quad (j = 1, 2, \ldots, n) \end{cases} \tag{3.2}$$

Prin utilizarea sistemului (3.2) se determină drumul de valoare minimă dintre x_1 şi x_j ţinînd seama de drumul de valoare minimă dintre x_1 şi x_i şi de $v(x_i, x_j)$ pentru $(x_i, x_j) \in U_{\bar{x}j}$. Aplicarea acestui procedeu presupune parcurgerea grafului de la stînga spre dreapta, iar folosirea algoritmului definit de punctele 1—3 este echivalentă cu aflarea drumului minim începînd de la nodul x_n către nodul x_1.

Scanned with OKEN Scanner

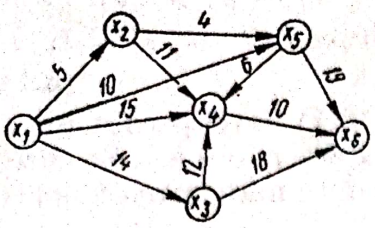


Fig. 3.5.

În cazul cînd se cere aflarea drumului de valoare maximă, valorile de start sînt $\lambda_1 = 0$ și $\lambda_j = -\infty$ $(j = 2, \ldots, n)$. Atunci cînd $\lambda_j - \lambda_i < v(x_i, x_j)$, valorile λ_j se înlocuiesc cu $\lambda'_j = \lambda_i + v(x_i, x_j)$. Dacă există mai multe variante care permit mărirea lui λ_i, atunci se alege indicatorul λ'_j cu valoarea maximă.

Aplicarea acestui algoritm este echivalentă cu rezolvarea sistemului:

$$\begin{cases} \lambda_1 = 0 \\ \lambda_j = \max\ [\lambda_i + v(x_i, x_j)] \quad (j = 1,2,\ldots,n] \\ (x_i, x_j) \in \mathbf{U}^-_{xj} \end{cases}$$

Trebuie precizat că este posibil ca graful să conțină mai multe drumuri de valoare minimă (maximă) și de aceea pentru determinarea acestora trebuie marcate toate succesiunile de arce (noduri) care conduc la aceeași valoare.

Exemplul 3.4. Pentru a ilustra tehnica de aplicare a algoritmului Ford vom rezolva o problemă de tipul celeia prezentate în exemplul 3.2. Considerăm graful dat în fig. 3.5. și ne propunem să determinăm drumul de valoare minimă între vîrfurile x_1 și x_6.

Rezolvarea problemei se poate face într-un tabel care conține în primele două coloane arcele grafului și valorile asociate acestora. În celelalte coloane sînt trecute diferențele $\lambda_j - \lambda_i$ calculate la fiecare iterație. Într-un alt tabel se trec valorile variabilelor λ_j, obținute după fiecare iterație (vezi tabelele 3.1 și 3.2.). Se constată că în tabelul 3.1. au fost trecute întîi arcele incidente spre interior lui x_6, apoi cele care au extremitatea finală x_5 ș.a.m.d.

• *Iterația 0* (start). Începem cu $\lambda_1 = 0$, $\lambda_2 = \lambda_3 = \ldots = \lambda_6 = \infty$.

• *Iterația 1.* Vom conveni ca $\infty - \infty = 0$! Efectuînd diferențele $\lambda_j - \lambda_i$ pentru toate arcele grafului dat, observăm că diferențele $\lambda_5 - \lambda_1$, $\lambda_4 - \lambda_1$, $\lambda_3 - \lambda_1$, și $\lambda_2 - \lambda_1$ sînt mai mari decît valorile arcelor corespunzătoare (ele au fost marcate în tabelul 3.1. printr-un asterisc).

Deoarece în iterația zero au rezultat diferențe $\lambda_j - \lambda_i > v(x_i, x_j)$ se aplică relația (3.1) și obținem:

$$\lambda_5^{(1)} = \lambda_1 + v(x_1, x_5) = 0 + 10 = 10$$
$$\lambda_4^{(1)} = \lambda_1 + v(x_1, x_4) = 0 + 15 = 15$$
$$\lambda_3^{(1)} = \lambda_1 + v(x_1, x_3) = 0 - 14 = 14$$
$$\lambda_2^{(1)} = \lambda_1 + v(x_1, x_2) = 0 + 5 = 5$$

Noile valori $\lambda_j^{(1)}$ se trec în tabelul 3.2. pe linia corespunzătoare primei iterații.

• *Iterația 2.* Se calculează diferențele $\lambda_j^{(1)} - \lambda_i^{(1)}$ și se compară cu $v(x_i, x_j)$, iar în cazul cînd $\lambda_j^{(1)} - \lambda_i^{(1)} > v(x_i, x_j)$ se calculează $\lambda_j^{(2)} = \lambda_i^{(1)} + v(x_i, x_j)$

Scanned with OKEN Scanner

Tabelul 3.1

(x_i, x_j)	$v(x_i, x_j)$	$\lambda_j^{(0)} - \lambda_i^{(0)}$	$\lambda_j^{(1)} - \lambda_i^{(1)}$	$\lambda_j^{(2)} - \lambda_i^{(2)}$
(x_5, x_6)	19	0	∞*	14
(x_4, x_6)	10	0	∞*	10 x_4
(x_3, x_6)	18	0	∞*	11
(x_2, x_5)	4	0	5*	4
(x_1, x_5)	10	∞*	10	9 x_2
(x_5, x_4)	6	0	5	6
(x_3, x_4)	12	0	1	1 x_5
(x_2, x_4)	11	0	10	10
(x_1, x_4)	15	∞*	15	15 x_1
(x_1, x_3)	14	∞*	14	14
(x_1, x_2)	5	∞*	5	5 x_1

Analizînd diferenţele $\lambda_j^{(1)} - \lambda_i^{(1)}$ se constată că diferenţele $\lambda_6^{(1)} - \lambda_5^{(1)}$, $\lambda_6^{(1)} - \lambda_4^{(1)}$, $\lambda_6^{(1)} - \lambda_3^{(1)}$ şi $\lambda_5^{(1)} - \lambda_2^{(1)}$ sînt mai mari decît valorile arcelor coresTabelul 3.2

Nr. iter.	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6
0	0	∞	∞	∞	∞	∞
1	0	5	14	15	10	∞
2	0	5	14	15	9	25

punzătoare şi de aceea ele s-au notat cu asterisc. Există trei variante pentru micşorarea lui λ_6:

$$\lambda_5^{(1)} + v(x_5, x_6) = 10 + 19 = 29$$

$$\lambda_4^{(1)} + v(x_4, x_6) = 15 + 10 = 25$$

$$\lambda_3^{(1)} + v(x_3, x_6) = 14 + 18 = 32.$$

Evident că în cazul problemelor de minim se alege valoarea cea mai mică, deci $\lambda_6^{(2)} = 25$. Ultima diferenţă impune micşorarea variabilei λ_5, deci $\lambda_5^{(2)} = \lambda_2^{(1)} + v(x_2, x_5) = 5 + 4 = 9$. Celelalte valori $\lambda_j^{(1)}$ se menţin nemodificate (vezi tabelul 3.2, iteraţia 2).

• *Iteraţia 3.* Se calculează $\lambda_j^{(2)} - \lambda_i^{(2)}$ constatîndu-se că nu mai există diferenţe pentru care $\lambda_j^{(2)} - \lambda_i^{(2)} > v(x_i, x_j)$, prin urmare valorile actuale $\lambda_j^{(2)}$ nu mai pot fi micşorate. Rezultă că valoarea minimă a drumului care uneşte x_1 cu x_6 este egală cu 25. Pentru stabilirea succesiunii de arce sau noduri, care formează drumul de valoare minimă se reţin arcele (nodurile) adiacente pentru care în ultima coloană a tabelului 3.1 s-au obţinut

$$\lambda_j^{(2)} - \lambda_i^{(2)} = v(x_i, x_j).$$

Scanned with OKEN Scanner

În acest fel s-au pus în evidență două drumuri de valoare minimă:

$$\mu_1 = (x_1,\ x_4,\ x_6) = 25$$
$$\mu_2 = (x_1,\ x_2,\ x_5,\ x_4,\ x_6) = 25$$

3.4.2. Algoritmul Bellman-Kalaba

Acest algoritm se bazează pe principiul general al programării dinamice, care în cazul de față capătă următoarea formulare: dacă drumul de valoare minimă (maximă) care unește x_1 cu x_n trece prin vîrfurile „intermediare" x_h și x_k atunci drumul parțial care unește $-x_h$ cu x_k este drumul de valoare minimă (maximă) între aceste două vîrfuri.

În cazul cînd se cere determinarea drumului de valoare minimă, grafului $\mathbf{G(X, \Gamma)}$ i se atașează matricea $\mathbf{C} = (c_{ij})$, $i, j = 1, \ldots, n$ în care elementele c_{ij} se definesc astfel:

$$c_{ij} = \begin{cases} v(x_i,\ x_j) & \text{dacă } (x_i,\ x_j) \in \mathbf{U} \quad (i \neq j) \\ \infty & \text{dacă } (x_i,\ x_j) \notin \mathbf{U}\ (i \neq j) \\ 0 & \text{dacă } i = j \end{cases}$$

Fiecărui vîrf x_i al grafului $\mathbf{G}$ i se asociază o variabilă v_i, care are semnificația de valoare a drumului care unește vîrful x_i cu x_n. Soluția problemei enunțate se obține prin rezolvarea sistemului de ecuații:

$$\begin{cases} v_i = \min\limits_{j \neq i} (v_j + c_{ij}), & i = 1, \ldots, n-1 \\ & j = 1, 2, \ldots, n \\ v_n = 0 \end{cases} \tag{3.3}$$

Într-adevăr se poate arăta că dacă $v_1, v_2, \ldots, v_n$ este soluția acestui sistem, atunci $\bar{v}_1$ reprezintă valoarea minimă a drumului care unește pe x_1 cu x_n.

Pentru rezolvarea sistemului (3.3) se utilizează un procedeu iterativ care pleacă de la informația inițială definită prin:

$$\begin{cases} v_i^{(0)} = c_{in} & i = 1,\ 2, \ldots, n-1 \\ v_n^{(0)} = 0 \end{cases}$$

Fiecare iterație k ($k = 1, 2 \ldots$) constă în rezolvarea sistemului:

$$\begin{cases} v_i^{(k)} = \min\limits_{j \neq i} (v_j^{(k-1)} + c_{ij}) & i = 1, \ldots, n-1 \\ & j = 1, 2, \ldots n \\ v_n^{(k)} = 0 & k = 1, 2, \ldots \end{cases} \tag{3.4}$$

Procesul de calcul se încheie în momentul în care se găsește:

$$v_i^{(k-1)} = v_i^{(k)} \quad i = 1, \ldots, n$$

Scanned with OKEN Scanner

Atunci $\bar{v}_i = v_i^{(k)}$, $(i = 1, \ldots, n)$ și valoarea minimă a drumului care unește x_1 cu x_n este $\bar{v}_1 = v_1^{(k)}$.

Determinarea arcelor drumului de valoare minimă $\bar{v}_1 = v_1^{(k)}$ se face astfel:

Fie x_{j_1} vîrful pentru care:

$$v_1^{(k)} = \min_{j \neq 1} (v_j^{k-1} + c_{1j}) = v_{j_1}^{(k-1)} + c_{1j_1}$$

Drumul de lungime minimă trecere prin x_{j_1} și conform principiului programării dinamice $y_{j_1}^{(k-1)}$ reprezintă valoarea minimă a drumului care unește x_{j_1} cu x_n. În continuare, să presupunem că:

$$v_{j_1}^{(k-1)} = \min_{j \neq j_1} (v_j^{(k-2)} + c_{j_1 j}) = v_{j_2}^{(k-2)} + c_{j_1 j_2}$$

Drumul μ căutat va trece prin x_{j_2}. Repetăm procedura plecînd de la $v_{j_2}^{(k-2)}$ ș.a.m.d. pînă cînd se ajunge la un $v_{j_k}^{(0)} = c_{j_k n}$.

Drumul de valoare minimă va fi:

$$\mu = (x_1, x_{j_1}, x_{j_2}, \ldots, x_{j_k}, x_n)$$

Este clar că dacă unul dintre vîrfurile intermediare nu este unic determinat vor exista mai multe drumuri de lungime minimă.

Calculele se pot organiza într-un tabel în care se trece matricea **C**, vectorul linie $v^{(0)} = (c_{1n}, c_{2n}, \ldots, c_{n-1,n}, 0)$ ale cărui componente sînt tocmai elementele ultimei coloane a matricei **C**, precum și vectorii linie $v^{(k)}$ ale căror elemente se calculează prin relațiile (3.4).

În vederea aplicării algoritmului Bellman — Kalaba pentru aflarea drumului de valoare maximă se introduc următoarele modificări.

a) La definirea elementelor matricei **C**, dacă $(x_i, x_j) \notin \mathbf{U}$ atunci $c_{ij} = -\infty$;

b) În relațiile (3.3) și (3.4) se înlocuiește operatorul de minim cu operatorul de maxim.

Exemplul 3.5. Pentru a ilustra tehnica de aplicare a algoritmului Bellman — Kalaba vom determina drumul de valoare maximă a grafului din figura 3.6 care s-a obținut prin concretizarea problemei formulate în exemplul 3.3.

Matricea asociată acestui graf este dată în tabelul 3.3. Acest tabel mai conține, pe lîngă elementele matricei **C**, vectorii linie $v_i^{(k)}$ $(k = 0, 1, 2, 3, 4, 5)$ calculați pe baza relației (3.4).

Se observă că vectorul v_i^0 s-a obținut prin transpunerea ultimei coloane a matricei **C**.

Pentru calcularea elementelor vectorului $v_i^{(1)}$ s-a efectuat suma între elementele vectorului $v_i^{(0)}$ și elementele corespunzătoare ale rîndului i, $(i=1, 2, \ldots, 7)$ a matricii **C** pentru $i \neq j$. Dintre sumele corespunzătoare fiecărui rînd s-a ales va-

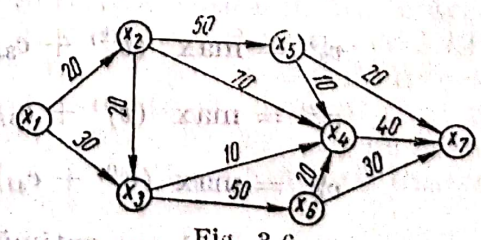


Fig. 3.6

loarea maximă. Spre exemplu :

$$v_2^{(1)} = \max(-\infty - \infty;\ 20 - \infty;\ 70 + 40;\ 50 + 20;\ -\infty + 30;\ -\infty + 0) = 110$$

$$v_3^{(1)} = \max(-\infty - \infty;\ -\infty - \infty;\ 10 + 40;\ -\infty + 20;\ 50 + 30;\ -\infty + 0) = 80$$

Tabelul 3.3.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	0	20	30	$-\infty$	$-\infty$	$-\infty$	$-\infty$
x_2	$-\infty$	0	20	70	50	$-\infty$	$-\infty$
x_3	$-\infty$	$-\infty$	0	10	$-\infty$	50	$-\infty$
x_4	$-\infty$	$-\infty$	$-\infty$	0	$-\infty$	$-\infty$	40
x_5	$-\infty$	$-\infty$	$-\infty$	10	0	$-\infty$	20
x_6	$-\infty$	$-\infty$	$-\infty$	20	$-\infty$	0	30
x_7	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	$-\infty$	0
$v_i^{(0)}$	$-\infty$	$-\infty$	$-\infty$	40	20	30	0
$v_i^{(1)}$	$-\infty$	110	80	40	50	60	0
$v_i^{(2)}$	130	110	110	40	50	60	0
$v_i^{(3)}$	140	130	110	40	50	60	0
$v_i^{(4)}$	150	130	110	40	50	60	0
$v_i^{(5)}$	150	130	110	40	50	60	0

În iteraţiile următoare se însumează elementele vectorului $v_i^{(1)}$ cu elementele corespunzătoare ale rîndului i ($i = 1, 2, \ldots, 7$) a matricei **C** reţinîndu-se pentru fiecare rînd valoarea maximă.

Întrucît $v_i^{(5)} = v_i^{(4)}$ rezultă că s-a obţinut soluţia optimă. Valoarea maximă a drumului de la x_1 la x_7 este $v_1^{(5)} = 150$.

Determinarea vîrfurilor prin care trece drumul de valoare optimă se face astfel :

$$v_1^{(5)} = \max_{j \neq 1} (v_j^{(4)} + c_{1j}) = v_2^{(4)} + c_{12} = 130 + 20 \qquad \boxed{x_2}$$

$$v_2^{(4)} = \max (v_j^{(3)} + c_{2j}) = v_3^{(3)} + c_{23} = 110 + 20 \qquad \boxed{x_3}$$

$$v_3^{(3)} = \max (v_j^{(2)} + c_{3j}) = v_6^{(2)} + c_{36} = 60 + 50 \qquad \boxed{x_6}$$

$$v_6^{(2)} = \max (v_j^{(1)} + c_{6j}) = v_4^{(1)} + c_{64} = 40 + 20 \qquad \boxed{x_4}$$

$$v_4^{(1)} = \max (v_j^{(0)} + c_{4j}) = v_7^{(0)} + c_{47} = 0 + 40 \qquad \boxed{x_7}$$

Deci drumul de valoare optimă este $\mu = (x_1, x_2, x_3, x_6, x_4, x_7)$

Scanned with OKEN Scanner

3.5. Rețele de transport

Exemplul 3.1 prezentat la începutul acestui capitol a pus în discuție o problemă economică care poate fi reprezentată printr-un anumit tip de graf, denumit rețea de transport.

Se numește rețea de transport un graf $\mathbf{G} = (\mathbf{X}, \mathbf{U})$ finit fără bucle cu următoarele proprietăți :

1) fiecărui arc $u \in \mathbf{U}$ îi este atașat un număr întreg $c(u) \geqslant 0$ care se numește capacitatea arcului u ;

2) există un singur vîrf x_0 caracterizat prin aplicația $\Gamma^{-1}(x_0) = \Phi$, ca urmare acesta nu este extremitate finală pentru nici un arc al grafului $\mathbf{G}$ și de aceea este numit intrare în rețea ;

3) există un singur vîrf x_n, caracterizat prin aplicația $\Gamma(x_n) = \Phi$, ca urmare acesta nu este extremitatea inițială pentru nici un arc al grafului $\mathbf{G}$ și de aceea este numit ieșire din rețea.

Vîrfurile $x_1, \ldots, x_{n-1}$ distincte de intrarea x_0 și ieșirea x_n se numesc vîrfuri intermediare.

O funcție φ definită pe mulțimea arcelor cu valori întregi ($\varphi : \mathbf{U} \to \mathbf{Z}$) se numește flux de rețeaua $\mathbf{G}(\mathbf{X}, \mathbf{U})$ dacă :

1) $0 \leqslant \varphi(u) \leqslant c(u)$ pentru orice arc $u \in \mathbf{U}$

2) pentru orice vîrf intermediar $x_i (1 \leqslant i \leqslant n-1)$ este îndeplinită condiția :

$$\sum_{u \in \mathbf{U}_{x_i}^-} \varphi(u) = \sum_{u \in \mathbf{U}_{x_i}^+} \varphi(u) \tag{3.5}$$

Asimilînd $\varphi(u)$ cu o cantitate de materie care parcurge arcul u în sensul orientării sale, condiția *1* exprimă cerința naturală ca această cantitate să nu depășească capacitatea arcului. Condiția *2* arată că în vîrfurile intermediare x_i, nu sînt „pierderi", în sensul că suma cantităților sosite în x_i de-a lungul arcelor incidente spre interior acestuia este egală cu suma cantităților care „ies" din x_i pe arcele incidente spre exterior lui.

Însumînd relațiile (3.5) pentru toate vîrfurile intermediare și reducînd termenii prezenți în ambii membri rezultă :

$$\sum_{u \in \mathbf{U}_{x_0}^+} \varphi(u) = \sum_{u \in \mathbf{U}_{x_n}^-} \varphi(u) = v(\varphi x_n) \tag{3.6}$$

Relația (3.6) arată că într-o rețea fără pierderi cantitatea de materie „intrată" în rețea prin x_0 este egală cu cantitatea sosită în x_n. Mărimea $v(\varphi x_n)$ se numește valoarea fluxului φ în vîrful x_n. Pentru astfel de probleme interesează determinarea fluxului de valoare maximă într-o rețea de transport dată. Se cunosc mai multe metode cu ajutorul cărora putem rezolva această problemă. Algoritmul Ford — Fulkerson, pe care îl vom prezenta în continuare, are la bază conceptul de tăietură într-o rețea și de aceea vom defini în prealabil această noțiune.

Scanned with OKEN Scanner

Vom considera o rețea de transport $\mathbf{G} = (\mathbf{X}, \mathbf{U})$ și o submulțime $\mathbf{A}$ de vîrfuri ale acesteia ($\mathbf{A} \subset \mathbf{X}$) care are proprietatea că $x_0 \notin \mathbf{A}$ și $x_n \in \mathbf{A}$.

Mulțimea $\mathbf{U}_{\mathbf{A}}^{-}$ a arcelor incidente spre interior acestei submulțimi se numește tăietura în rețeaua $\mathbf{G}$. Capacitatea tăieturii $\mathbf{U}_{\mathbf{A}}^{-}$ se definește ca fiind suma capacităților arcelor componente:

$$\mathbf{C}(\mathbf{U}_{\mathbf{A}}^{-}) = \sum_{u \in \mathbf{U}_{\mathbf{A}}^{-}} c(u)$$

Este ușor de văzut că o unitate de cantitate de materie care pleacă din x_0 străbate cu necesitate un arc al tăieturii $\mathbf{U}_{\mathbf{A}}^{-}$. Se deduce imediat că pentru orice flux φ și orice tăietură $\mathbf{U}_{\mathbf{A}}^{-}$ avem:

$$v(\varphi_{x_n}) \leqslant \mathbf{C}(\mathbf{U}_{\mathbf{A}}^{-})$$

Teorema Ford—Fulkerson. Într-o rețea de transport un flux are valoarea maximă dacă și numai dacă există o tăietură de capacitate egală cu valoarea fluxului. În acest caz tăietura are capacitatea minimă.

Deci, în cazul egalității între flux și tăietură, $v(\varphi_{x_n}) = \mathbf{C}(\mathbf{U}_{\mathbf{A}}^{-})$, atunci fluxul φ_{x_n} are valoarea maximă în timp ce tăietura $\mathbf{U}_{\mathbf{A}}^{-}$ are capacitatea minimă. Acest rezultat constituie condiția necesară și suficientă de existență a fluxului de valoare maximă și stă la baza algoritmului Ford—Fulkerson pentru determinarea fluxului de valoare maximă. Aplicarea acestui algoritm constă în următoarele:

1) Se propagă în rețea, din vîrful x_0 către x_n, un flux φ astfel încît $\varphi(u) \leqslant c(u)$;

2) Se marchează vîrfurile rețelei $\mathbf{G}$ după următoarea regulă:

2a) Intrarea x_0 se marchează cu $[+]$;

2b) dacă vîrful x_j este nemarcat și există un arc (x_i, x_j) nesaturat a cărui extremitate inițială x_i este marcată atunci marcăm x_j cu $[+ x_i]$. Precizăm că în cazul cînd arcul (x_i, x_j) este saturat, adică $\varphi(u) = c(u)$, nu se poate marca vîrful x_j.

2c) dacă vîrful x_j este nemarcat și există arcul (x_j, x_i) străbătut de un flux nenul a cărui extremitate finală x_i este marcată, atunci marcăm x_j cu $[-x_i]$, iar în cazul că fluxul este nul nu se marchează vîrful x_j. Dacă după aplicarea procedurii de marcare ieșirea x_n nu este marcată atunci fluxul φ are valoarea maximă. Într-adevăr fie $\mathbf{A}$ mulțimea vîrfurilor nemarcate; deoarece x_0 este marcat iar x_n nemarcat, $\mathbf{A}$ definește o tăietură $\mathbf{U}_{\mathbf{A}}^{-}$ ale cărei arce sînt saturate în virtutea regulii de marcare *2b*. Însumînd egalitățile (3.5) pentru toate punctele intermediare x_i din $\mathbf{A}$ obținem:

$$\sum_{u \in \mathbf{U}_{\mathbf{A}}^{-}} \varphi(u) = \sum_{u \in \mathbf{U}_{\mathbf{A}}^{+}} \varphi(u) + v(\varphi)$$

dar $\sum_{u \in \mathbf{U}_{\mathbf{A}}^{-}} \varphi(u) = \sum_{u \in \mathbf{U}_{\mathbf{A}}^{-}} c(u) = \mathbf{C}(\mathbf{U}_{\mathbf{A}}^{-})$ deoarece arcele tăieturii sînt saturate iar $\sum_{u \in \mathbf{U}_{\mathbf{A}}^{+}} \varphi(u) = 0$ pentru orice arc u incident spre exterior mul-

ţimii A (în virtutea regulii 2c de marcare). Prin urmare

$$v(\varphi x_n) = C(U_A^-)$$

şi conform teoremei Ford — Fulkerson, φx_n are valoarea maximă.

În cazul în care ieşirea x_n este marcată va exista un lanţ γ de arce cu extremităţile marcate care uneşte intrarea x_0 cu ieşirea x_n.
Să notăm cu

$$d_\gamma^1 = \min_{u \in U_\gamma^{\rightarrow}} [c(u) - \varphi(u)]$$

$U_\gamma^{\rightarrow}$ fiind mulţimea arcelor lanţului γ a căror extremitate finală este marcată cu [+] (se face abstracţie de vîrful în raport cu care s-a făcut marcarea), prin urmare mulţimea $U_\gamma^{\rightarrow}$ este formată din arcele orientate în sensul „intrare-ieşire". Fie

$$d_\gamma^2 = \min_{u \in U_\gamma^{\leftarrow}} \varphi(u)$$

$U_\gamma^{\leftarrow}$ fiind mulţimea arcelor lanţului γ ale căror extremităţi iniţiale sînt notate cu [—]. Acestea sînt arcele orientate invers sensului „intrare-ieşire".

Indicatorul $d_\gamma = \min(d_\gamma^1, d_\gamma^2) > 0$. (Dacă mulţimea $U_\gamma^{\leftarrow}$ este vidă atunci $d_\gamma^2 = 0$).

Adăugînd mărimea d_γ la fluxurile arcelor din $U_\gamma^{\rightarrow}$, scăzînd aceeaşi mărime din fluxurile arcelor mulţimii $U_\gamma^{\leftarrow}$ şi menţinînd neschimbate fluxurile arcelor ce nu fac parte din γ obţinem un nou flux φ' „mai bun" decît φ, deoarece

$$v(\varphi') = v(\varphi) + d_\gamma > v(\varphi)$$

În esenţă algoritmul de determinare a fluxului de valoare maximă constă în îmbunătăţirea succesivă a unui flux iniţial dat cu ajutorul procedurii de marcare descrisă mai sus. Cum această procedură este destul de laborioasă, se evită utilizarea ei, în toate iteraţiile căutîndu-se mai întîi un flux complet, adică un flux cu proprietatea că orice drum care uneşte intrarea cu ieşirea din reţea să conţină cel puţin un arc saturat. Dacă la o anumită iteraţie dispunem de un flux care nu este complet, va exista un drum γ — care uneşte intrarea cu ieşirea din reţea, compus numai din arce nesaturate. Dacă notăm :

$$\bar{d}\gamma = \min_{u \in \gamma} [c(u) - \varphi(u)] > 0$$

atunci mărind fluxurile arcelor drumului γ cu mărimea $\bar{d}\gamma$ şi menţinînd neschimbate fluxurile celorlalte arce care nu fac parte din γ, obţinem un flux φ' „mai bun" decît φ, în sensul că : $v(\varphi') = v(\varphi) + \bar{d}_r > v(\varphi)$. În plus drumul γ va conţine acum cel puţin un arc saturat. Într-o a doua etapă aplicăm fluxului complet procedura de etichetare. Dacă fluxul curent φ nu are valoarea maximă, aceasta se poate mări de-a lungul unui lanţ de arce care uneşte intrarea cu ieşirea reţelei, aşa cum am văzut mai sus. Noului flux i se aplică procedura de etichetare ş.a.m.d.

Scanned with OKEN Scanner

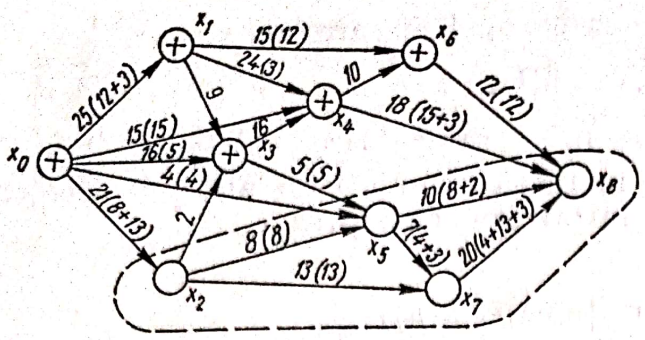


Fig. 3.7.

Algoritmul se termină în momentul în care ieşirea reţelei nu mai poate fi marcată.

Exemplul 3.6. Considerăm reţeaua de transport dată în figura 3.7 şi ne propunem să determinăm fluxul maxim ce poate fi transmis de la vîrful x_0 la x_8.

Prima operaţie constă în propagarea unui flux $\varphi(u)$ pe fiecare arc astfel încît să nu fie depăşită capacitatea arcelor. Aceste fluxuri sînt înregistrate în paranteză alături de capacităţile corespunzătoare fiecărui arc. Astfel, spre exemplu, pe drumurile $(x_0, x_3, x_5, x_7, x_8)$ şi (x_0, x_3, x_5, x_8) s-a propagat un flux de 5 unităţi pentru a nu depăşi capacitatea arcului (x_3, x_5).

Se constată că după efectuarea operaţiei de propagare a fluxurilor, toate fluxurile înregistrate pe drumurile ce au ca nod iniţial pe x_0 şi ca nod final pe x_8 sînt complete avînd cel puţin un arc saturat.

Folosind procedura de marcare descrisă mai înainte se constată că vîrful x_8, care este ieşirea reţelei, nu se poate marca, deci s-a obţinut fluxul maxim. De asemenea nu sînt marcate vîrfurile : x_2, x_5, x_7. Mulţimea $\mathbf{A} = (x_2, x_5, x_7, x_8)$ defineşte o tăietură a cărei capacitate este :

$$\mathbf{C}(\mathbf{U}_{\mathbf{A}}^{-}) = \sum_{u \in \mathbf{U}_{\mathbf{A}}^{-}} c(u) = 21 + 4 + 5 + 18 + 12 = 60$$

Conform teoremei Ford—Fulkerson, fluxul are valoarea maximă atunci cînd $v(\varphi_{x_8}) = \mathbf{C}(\mathbf{U}_{\mathbf{A}}^{-})$. Se constată că $v(\varphi_{x_8}) = 12 + 18 + 10 + 20 = 60$ deci φ_{x_8} are valoare maximă.

3.6. Drumuri şi circuite hamiltoniene

3.6.1. Definirea problemei

Într-un graf $\mathbf{G} = (\mathbf{X}, \Gamma)$, un drum $\mu = (x_{i_1}, x_{i_2}, \ldots, x_{i_n})$ se numeşte „hamiltonian" dacă trece o dată şi numai o dată prin fiecare vîrf al grafului. Dacă graful $\mathbf{G}$ de ordinul n are drumuri hamiltoniene, lungimea acestora este $l(\mu) = n - 1$.

Rezultă că drumurile hamiltoniene sînt drumuri elementare de lungime $(n - 1)$.

Pe baza unor modele matematice adecvate se pot determina drumurile hamiltoniene într-un graf ataşat unei probleme economice. Astfel, serviciul poştal al unui oraş are sarcina să asigure ridicarea corespondenţei de la toate punctele unde sînt amplasate cutii poştale. O cale eficientă de

Scanned with OKEN Scanner

rezolvare a acestei probleme constă în determinarea drumurilor hamiltoniene într-un graf $\mathbf{G}$ definit prin mulțimea $\mathbf{X}$ a punctelor unde sînt amplasate cutiile poștale și prin aplicațiile $\mathbf{\Gamma}$ care reprezintă rețeaua de drumuri ce unesc punctele $x_i \in \mathbf{X}$. Dacă există mai multe drumuri hamiltoniene se pune problema alegerii drumului de valoare optimă folosind drept criteriu de eficiență economică costul transportului, durata transportului etc. În acest scop se determină valoarea fiecărui drum μ prin însumarea indicatorilor $v(x_i, x_j)$ atașați arcelor ce compun drumul respectiv și în final se alege varianta optimă. Domeniul de aplicare al acestui tip de model economico-matematic poate fi extins la soluționarea unor variate probleme economice de planificare și organizare.

Un alt concept caracteristic problemei formulate mai sus este circuitul hamiltonian. Circuitul $\mu = (x_{i1}, x_{i2}, \ldots, x_{in}, x_{i1})$ este hamiltonian dacă trece o dată și numai o dată prin fiecare vîrf al grafului cu excepția vîrfului x_{i1}. Dacă graful este de ordinul n, lungimea circuitului hamiltonian este $l(\mu) = n$. Prin urmare circuitele hamiltoniene constituie clasa circuitelor elementare de lungime „n".

O condiție necesară pentru ca să existe cel puțin un circuit hamiltonian este ca graful să fie tare conex. O condiție necesară pentru existența unui drum hamiltonian este ca graful să fie complet.

Dacă graful este neorientat atunci analogul drumului sau circuitului hamiltonian este lanțul, respectiv circuitul hamiltonian.

3.6.2. Metode de aflare a drumurilor hamiltoniene

Există mai multe metode pentru aflarea drumurilor și circuitelor hamiltoniene. În continuare vom prezenta metoda înmulțirii latine și metoda Foulkes.

3.6.2.1. *Metoda înmulțirii latine*

În cadrul acestei metode se operează cu produse matriceale de un tip special care oferă posibilitatea enumerării succesive fără redondanță, a tuturor drumurilor de lungime $1, 2, \ldots, n-1$.

La baza calculelor ce constituie algoritmul înmulțirii latine stă matricea $\mathbf{M}^{(1)}$ formată din elemente m_{ij} definite astfel:

$$m_{ij} = \begin{cases} x_i x_j & \text{dacă arcul } (x_i x_j) \in \mathbf{U}; \\ 0 & \text{dacă nu există arc de la } x_i \text{ la } x_j \text{ și pentru } i = j. \end{cases}$$

Din această matrice latină se deduce matricea $\tilde{\mathbf{M}}^{(1)}$ suprimînd nodul inițial (prima literă) la fiecare element m_{ij}.

Prin înmulțirea latină a matricei $\mathbf{M}^{(1)}$ cu matricea $\tilde{\mathbf{M}}^{(1)}$ se obține matricea $\mathbf{M}^{(2)}$.

Operația de înmulțire se efectuează ca în cazul înmulțirii clasice „linie cu coloană", dar în loc să efectuăm înmulțire (ceea ce aici este lipsit de

Scanned with OKEN Scanner

sens) cînd o căsuţă ce corespunde elementului m_{ij} trebuie înmulţită cu o căsuţă (j, k), atunci în căsuţa (i, k) a matricii $\mathbf{M}^{(2)}$ se va înscrie :

a) cifra *0* — dacă pentru toţi j una şi cealaltă sau numai una dintre aceste căsuţe conţine *zero* ;

b) cifra *0* — dacă luînd produsul „linie cu coloană pentru toate căsuţele examinate, punînd după fiecare succesiune dintre căsuţa (i, j) a lui $\mathbf{M}^{(1)}$ o succesiune din căsuţa (j, k) a lui $\widetilde{\mathbf{M}}^{(1)}$, nu se poate forma pentru toţi j o succesiune care să nu conţină o literă repetată ;

c) toate succesiunile distincte obţinute prin juxtapunere la succesiunea din căsuţa (i, j) a matricii $\mathbf{M}^{(1)}$, a unei succesiuni din căsuţa (j, k) a matricii $\widetilde{\mathbf{M}}^{(1)}$.

În felul acesta dacă notăm cu **L** produsul latin se obţine :

$$\mathbf{M}^{(1)}\mathbf{L}\,\widetilde{\mathbf{M}}^{(1)} = \mathbf{M}^{(2)}$$

Matricea $\mathbf{M}^{(1)}$ dă lista drumurilor elementare de lungime *1*, iar matricea $\mathbf{M}^{(2)}$ dă lista drumurilor elementare de lungime *2*.

Folosind regulile de înmulţire latină prezentate mai înainte se calculează matricea $\mathbf{M}^{(3)}$:

$$\mathbf{M}^{(3)} = \mathbf{M}^{(2)}\mathbf{L}\,\widetilde{\mathbf{M}}^{(1)}$$

care ne dă lista tuturor drumurilor elementare de lungime *3*. În continuare se calculează :

$$\mathbf{M}^{(3)}\mathbf{L}\,\widetilde{\mathbf{M}}^{(1)} = \mathbf{M}^{(4)}$$

. . . . . .

$$\mathbf{M}^{(k)}\mathbf{L}\,\widetilde{\mathbf{M}}^{(1)} = \mathbf{M}^{(k+1)}$$

Din ultima „matrice" obţinem lista tuturor drumurilor elementare de lungime $(k + 1)$.

Dacă ordinul grafului „n" este mare, atunci avem de efectuat un volum mare de „înmulţiri" de acest tip.

Aceste calcule pot fi scurtate în mod apreciabil, deoarece înmulţirea latină poate fi extinsă la matricea $\widetilde{\mathbf{M}}^{(s)}$ $(S > 1)$ în modul următor :

$$\mathbf{M}^{(r)}\mathbf{L}\,\widetilde{\mathbf{M}}^{(s)} = \mathbf{M}^{(r+s)}$$

unde $\widetilde{\mathbf{M}}^{(s)}$ se obţine din matricea $\mathbf{M}^{(s)}$ suprimînd prima literă din fiecare succesiune ce figurează în matrice.

Aşadar în ipoteza că nu ne interesează drumurile elementare mai mici decît n, va fi suficient să se calculeze succesiv : $\mathbf{M}^{(2)}$; $\mathbf{M}^{(4)} = \mathbf{M}^{(2)}\,\mathbf{L}\,\widetilde{\mathbf{M}}^{(2)}$; $\mathbf{M}^{(2^3)} = \mathbf{M}^{(4)}\,\mathbf{L}\,\widetilde{\mathbf{M}}^{(4)}$ şi $\mathbf{M}^{(2^k)} = \mathbf{M}^{(2^{k-1})}\,\mathbf{L}\,\widetilde{\mathbf{M}}^{(2^{k-1})}$ cu $2^k \leqslant n$. Prin organizarea corespunzătoare a produselor ce definesc matricea $\mathbf{M}^{(n)}$, volumul calculelor se poate reduce şi mai mult. Spre exemplu pentru a determina matricele $\mathbf{M}^{(33)}$ şi $\mathbf{M}^{(34)}$ se calculează : $\mathbf{M}^{(2)}$, $\mathbf{M}^{(4)}$, $\mathbf{M}^{(8)}$, $\mathbf{M}^{(16)}$ $\widetilde{\mathbf{M}}^{(32)}$, $\mathbf{M}^{(33)} = \mathbf{M}^{(32)}\mathbf{L}\,\widetilde{\mathbf{M}}^{(1)}$ şi $\mathbf{M}^{(34)} = \mathbf{M}^{(32)}\,\mathbf{L}\,\widetilde{\mathbf{M}}^{(2)}$.

Scanned with OKEN Scanner

Metoda „înmulţirii latine „poate fi programată uşor pe un calculator electronic şi dacă se modifică adecvat ea ne permite să găsim drumuri cărora li s-au impus restricţii suplimentare.

Pentru a determina circuitele hamiltoniene (dacă există), din matricea $\mathbf{M}^{(n-1)}$ deducem matricea $\widetilde{\mathbf{M}}^{(n-1)}$ după regula obişnuită şi calculăm :

$$\mathbf{M}^{*}_{(n)} = \widetilde{\mathbf{M}}^{(n-1)} \, \mathbf{L} \, \widetilde{\mathbf{M}}^{(1)}$$

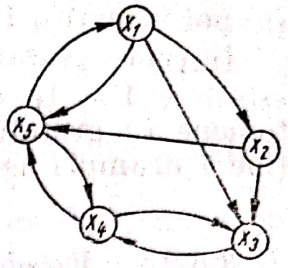


Fig. 3.8.

Prin adăugarea la începutul fiecărei succesiuni latine din matricea $\mathbf{M}^{*}_{(n)}$ a literei ce constituie indicele liniei acestei matrice se obţin circuitele hamiltoniene (dacă există).

Dacă graful este de ordin mare şi nu este tare conex, pentru a determina drumurile hamiltoniene este preferabil să-l descompunem în subgrafe tare conexe. În acest caz se foloseşte algoritmul Foulkes (vezi secţiunea 3.6.2.2).

Exemplul 3.7. În acest exemplu vom aplica metoda înmulţirii latine pentru calculul drumurilor hamiltoniene a grafului dată în figura 3.8.

Matricele $\mathbf{M}^{(1)}$ şi $\widetilde{\mathbf{M}}^{(1)}$ ataşate acestui graf au structura :

$$\mathbf{M}^{(1)} = \begin{pmatrix} 0 & x_1x_2 & x_1x_3 & 0 & x_2x_5 \\ 0 & 0 & x_2x_3 & 0 & x_2x_5 \\ 0 & 0 & 0 & x_3x_4 & 0 \\ 0 & 0 & x_4x_3 & 0 & x_4x_5 \\ x_5x_1 & 0 & 0 & x_5x_4 & 0 \end{pmatrix} ; \quad \widetilde{\mathbf{M}}^{(1)} = \begin{pmatrix} 0 & x_2 & x_3 & 0 & x_5 \\ 0 & 0 & x_3 & 0 & x_5 \\ 0 & 0 & 0 & x_4 & 0 \\ 0 & 0 & x_3 & 0 & x_5 \\ x_1 & 0 & 0 & x_4 & 0 \end{pmatrix}$$

Efectuînd înmulţirea latină a acestor două matrice se obţine matricea $\mathbf{M}^{(2)}$:

$$\mathbf{M}^{(2)} = \mathbf{M}^{(1)} \, \mathbf{L} \, \widetilde{\mathbf{M}}^{(1)} = \begin{pmatrix} 0 & 0 & x_1x_2x_3 & x_1x_3x_4 & x_1x_2x_5 \\ & & & x_1x_5x_4 & \\ x_2x_5x_1 & 0 & 0 & x_2x_3x_4 & 0 \\ & & & x_2x_5x_4 & \\ 0 & 0 & 0 & 0 & x_3x_4x_5 \\ x_4x_5x_1 & 0 & 0 & 0 & 0 \\ 0 & x_5x_1x_2 & x_5x_1x_3 & 0 & 0 \\ & & x_5x_4x_3 & & \end{pmatrix}$$

Matricea $\mathbf{M}^{(2)}$ dă lista drumurilor elementare de lungime 2. Eliminînd din fiecare succesiune a acestei matrice prima literă se obţine $\widetilde{\mathbf{M}}^{(2)}$ care se utilizează pentru a calcula matricea $\mathbf{M}^{(4)}$:

$$\mathbf{M}^{(4)} = \mathbf{M}^{(2)} \mathbf{L} \, \widetilde{\mathbf{M}}^{(2)} = \begin{pmatrix} 0 & 0 & x_1x_2x_5x_4x_3 & 0 & x_1x_2x_3x_4x_5 \\ x_2x_3x_4x_5x_1 & 0 & 0 & x_2x_5x_1x_3x_4 & 0 \\ 0 & x_3x_4x_5x_1x_2 & 0 & 0 & 0 \\ 0 & 0 & x_4x_5x_1x_2x_3 & 0 & 0 \\ 0 & 0 & 0 & x_5x_1x_2x_3x_4 & 0 \end{pmatrix}$$

Scanned with OKEN Scanner

Același rezultat se obține prin calcularea matricei $\mathbf{M}^{(3)} = \mathbf{M}^{(2)}\mathbf{L}\,\widetilde{\mathbf{M}}^{(1)}$ și apoi a matricei $\mathbf{M}^{(4)} = \mathbf{M}^{(3)}\mathbf{L}\,\widetilde{\mathbf{M}}^{(1)}$.

Întrucît graful are 5 vîrfuri și lungimea drumurilor hamiltoniene este $n - 1 = 4$, rezultă că matricea $\mathbf{M}^{(4)}$ conține toate drumurile hamiltoniene ale grafului considerat. Se constată că graful din figura 3.8 conține 7 drumuri hamiltoniene.

3.6.2.2. *Metoda lui Foulkes*

După cum s-a arătat în paragraful 3.2, oricărui graf $\mathbf{G} = (\mathbf{X}, \mathbf{U})$ i se poate asocia o matrice booleană $\mathbf{M}$. Această matrice stă la baza algoritmului Foulkes.

Dacă considerăm acum un graf finit $\mathbf{G}$ cu vîrfurile $x_1, \ldots, x_n$ și matricea booleană $\mathbf{M}$ asociată acestuia, atunci prin înmulțire booleană * putem determina matricea $\mathbf{M}^{(2)}$:

$$\mathbf{M}^{(2)} = \mathbf{M} \dot{\times} \mathbf{M} = \| m_{ij}^{(2)} \| \; (i, j, = 1, \ldots, n).$$

unde conform definiției $m_{ij}^{(2)} = m_{i1} \dot{\times} m_{1j} \dot{+} m_{i2} \dot{\times} m_{2j} \dot{+} \ldots \dot{+} m_{in} \dot{\times} m_{nj}$. Elementul $m_{ij}^{(2)}$ va fi 1 dacă cel puțin unul din termenii ce-l definesc este 1, fie acela $m_{ik} \dot{\times} m_{kj}$ $(1 \leqslant k \leqslant n)$. Produsul $m_{ik} \dot{\times} m_{kj} = 1$, dacă și numai dacă $m_{ik} = 1$, și $m_{kj} = 1$, adică există un arc de la x_i la x_k și un arc de la x_k la x_j.

Aceasta înseamnă că există un drum de lungime 2 de la x_i la x_j. Rezultă că $m_{ij}^{(2)} = 1$ dacă și numai dacă există cel puțin un drum de lungime 2 de la x_i la x_j. Prin urmare matricea $\mathbf{M}^{(2)}$ oferă informații asupra existenței drumurilor de lungime 2 între diferitele vîrfuri ale grafului $\mathbf{G}$. Inductiv, matricea $\mathbf{M}^{(r)}$ oferă informații asupra existenței drumurilor de lungime r, între diferitele vîrfuri ale grafului.

De aici rezultă imediat că elementele matricei $(\mathbf{I} \dot{+} \mathbf{M})^{(r)}$ (arată dacă între două vîrfuri oarecare ale grafului $\mathbf{G}$ există sau nu un drum de lungime mai mică sau egală cu r. Calculul matricei $(\mathbf{I} \dot{+} \mathbf{M})^{(r)}$ se face cu ajutorul relației :

$$(\mathbf{I} \dot{+} \mathbf{M})^{(r)} = \mathbf{I} \dot{+} \mathbf{M}^{(1)} \dot{+} \mathbf{M}^{(2)} \dot{+} \ldots \dot{+} \mathbf{M}^{(r)} \qquad (3.7)$$

Elementul $d_{ij}^{(r)}$ al acestei matrice poate lua următoarea valoare :

$$d_{ij}^{(r)} = \begin{cases} 1 \text{ dacă } x_j \in \{x_i\} \cup \Gamma(x_i) \cup \Gamma^2(x_i) \cup \ldots \cup \Gamma^r(x_i) \\ 0 \text{ în rest} \end{cases}$$

---

* Adunarea booleană se notează cu semnul $\dot{+}$ iar înmulțirea booleană cu semnul $\dot{\times}$. Acestea se definesc astfel : $0 \dot{+} 0 = 0$, $1 \dot{+} 0 = 0 \dot{+} 1 = 1 \dot{+} 1 = 1$ și respectiv $1 \dot{\times} 1 = 1$; $1 \dot{\times} 0 = 0 \dot{\times} 1 = 0 \dot{\times} 0 = 0$. Prin urmare dacă considerăm două matrice booleene de ordinul n, $\mathbf{A} = (a_{ij})$ și $\mathbf{B} = (b_{ij})$, atunci suma lor se definește astfel : $\mathbf{A} \dot{+} \mathbf{B} = (a_{ij} \dot{+} b_{ij})$. Produsul boolean al matricelor $\mathbf{A}$ și $\mathbf{B}$ este matricea $\mathbf{P} = (p_{ij})$ ale cărei elemente se definesc astfel :

$$p_{ij} = a_{i1} \dot{\times} b_{1j} \dot{+} a_{i2} \dot{\times} b_{2j} \dot{+} \ldots \dot{+} a_{in} \dot{\times} b_{nj} \; (i, j = 1,2,\ldots,n).$$

Scanned with OKEN Scanner

Mulțimea $\hat{\Gamma}(x_i) = \{x_i\} \cup \Gamma(x_i) \cup \Gamma^2(x_i) \cup \ldots \cup \Gamma^r(x_i)$ se numește închidere tranzitivă * a lui Γ, deci $\hat{\Gamma}(x_i)$ este o aplicație a mulțimii $\mathbf{X}$ în $\mathbf{X}$ definită prin relația precedentă. Avînd în vedere că $\Gamma^2(x_i) = \Gamma[\Gamma(x_i)]$, $\Gamma^3(x_i) = \Gamma\,[\Gamma^2(x_i)]$ etc., rezultă că $x_j \in \hat{\Gamma}(x_i)$ arată că între vîrfurile x_i și x_j există un drum de lungime mai mică sau egală cu r.

Pentru un graf cu n vîrfuri orice drum elementar este de lungime cel mult egală cu $n-1$, deci matricea $(\mathbf{I} \dot{+} \mathbf{M})^{(n-1)}$ va conține toate drumurile elementare sau neelementare de lungime mai mică sau egală cu $n-1$.

Întrucît orice drum este o reuniune de drumuri elementare matricea $(\mathbf{I} \dot{+} \mathbf{M})^{(n+p-1)}$ este în mod necesar egală cu $(\mathbf{I} \dot{+} \mathbf{M})^{(n-1)}$, deci:

$$(\mathbf{I} \dot{+} \mathbf{M})^{(n+p-1)} = (\mathbf{I} \dot{+} \mathbf{M})^{(n-1)} \quad p \geqslant 1$$

Prin urmare este suficient să calculăm $(\mathbf{I} \dot{+} \mathbf{M})^{(n-1)}$ pentru a afla dacă există un drum între orice vîrfuri x_i și x_j al grafului $\mathbf{G}$. Pentru reducerea calculelor se recomandă calcularea succesivă a matricelor: $(\mathbf{I} \dot{+} \mathbf{M})^{(2)}$, $(\mathbf{I} \dot{+} \mathbf{M})^{(2^2)}, \ldots, (\mathbf{I} \dot{+} \mathbf{M})^{(2^k)}$. Operația de înmulțire booleană se oprește atunci cînd

$$(\mathbf{I} \dot{+} \mathbf{M})^{(2^{k+1})} = (\mathbf{I} \dot{+} \mathbf{M})^{(2^k)} \tag{3.8}$$

sau cînd $2^k \geqslant n-1$.

După obținerea informațiilor cu privire la existența drumurilor hamiltoniene între două vîrfuri oarecare x_i și x_j se pune problema găsirii succesiunii de arce care formează fiecare drum. În acest scop graful $\mathbf{G}$ se descompune în clase de echivalență.

O mulțime de elemente $\mathbf{C}$ constituie o clasă de echivalență dacă elementele sale satisfac proprietățile de reflexivitate, simetrie și tranzitivitate care se definesc astfel:

a) Pentru orice $x_i \in \mathbf{C}$ există relația de echivalență $x_i \sim x_i$ (reflexivitate).

b) Dacă pentru oricare două vîrfuri $x_i, x_j \in \mathbf{C}$ există relația de echivalență $x_i \sim x_j$, atunci există și relația de echivalență $x_j \sim x_i$ (simetrie).

c) Dacă pentru vîrfurile $x_i, x_j, x_k \in \mathbf{C}$ există relațiile de echivalență $x_i \sim x_j$ și $x_j \sim x_k$, atunci există și relația de echivalență $x_i \sim x_k$ (tranzitivitate).

Precizăm că relația $x_i \sim x_j$ se interpretează: „x_i este echivalent cu x_j".

Fiind dat un graf $\mathbf{G}(\mathbf{X}, \Gamma)$ vom considera că între vîrfurile $x_i \in \mathbf{X}$ și $x_j \in \mathbf{X}$ există relația de echivalență $x_i \sim x_j$ dacă și numai dacă există un drum de la x_i la x_j și invers și prin urmare trebuie satisfăcute relațiile:

$$x_j \in \hat{\Gamma}(x_i) \quad \text{și} \quad x_i \in \hat{\Gamma}(x_j)$$

---

* Dacă considerăm o mulțime $\mathbf{X}$ de persoane iar $x_i \in \mathbf{X}$ o submulțime a acestuia, atunci $\Gamma(x_i)$ poate fi mulțimea copiilor lui x_i, $\Gamma^2(x_i)$ este mulțimea nepoților lui x_i, iar $\hat{\Gamma}(x_i) = \{x_i\} \cup \Gamma(x_i) \cup \Gamma^2(x_i)$ reprezintă mulțimea formată din x_i și a descendenților.

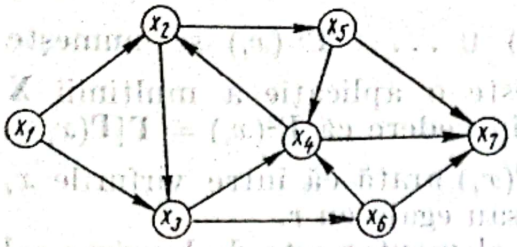


Fig. 3.9.

Aşadar, o relaţie de echivalenţă astfel definită va determina un subgraf tare conex al grafului **G**.

Dacă vîrfurile x_i şi x_j aparţin la două clase de echivalenţă şi există un drum de la x_i şi x_j ($x_j \in \hat{\Gamma}(x_i)$) atunci nu există un drum de la x_j la x_i (dacă ar exista atunci $x_i \sim x_j$).

Considerînd că mulţimea **C** este clasa de echivalenţă la care aparţine x_i şi **C'** clasa de echivalenţă a lui x_j, atunci va exista un drum de la fiecare vîrf a lui **C** la fiecare vîrf a lui **C'** şi nu există nici un drum de la oricare vîrf a lui **C'** la un vîrf a lui **C**. În acest fel s-a definit o relaţie de ordine tranzitivă şi antisimetrică *.

Pentru a obţine uşor vîrfurile care aparţin aceleiaşi clase de echivalenţă este suficient să remarcăm că dacă două vîrfuri aparţin aceleiaşi clase, atunci în matricea $(\mathbf{I} \dot{+} \mathbf{M})^{(2^k)}$ liniile corespunzătoare celor două vîrfuri sînt identice. Rezultă că prin ordonarea liniilor (vîrfurilor) ţinînd seama de numărul de elemente egale cu 1 se obţin clasele de echivalenţă ale grafului **G**. După această operaţie matricea capătă un aspect semitriunghiular. Dacă matricea astfel ordonată este strict triunghiulară în raport cu submatricele referitoare la clasele de echivalenţă, atunci există ordine totală şi pot exista unul sau mai multe drumuri hamiltoniene. În caz contrar nu există nici un drum hamiltonian.

Enumerarea drumurilor hamiltoniene se face luînd întîi în consideraţie drumurile hamiltoniene existente în cadrul fiecărei clase de echivalenţă, apoi acestea se racordează la arcele care fac parte din drumurile care traversează toate clasele.

Exemplul 3.8. Să se determine drumurile hamiltoniene ale grafului dat în figura 3.9.

Matricea **M** asociată grafului din figura 3.9 are următoarea structură:

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Conform relaţiilor (3.7) şi (3.8) trebuie să calculăm: $(\mathbf{I} \dot{+} \mathbf{M})^{(2^k)} = \mathbf{I} \dot{+} \mathbf{M}^{(1)} \dot{+} \mathbf{M}^{(2)} \dot{+} \ldots \dot{+} \mathbf{M}^{(2^k)}$ cu $2^k \geqslant 6$. Pentru reducerea numărului de operaţii vom calcula succesiv matricele: $(\mathbf{I} \dot{+} \mathbf{M})^{(2)}$, $(\mathbf{I} \dot{+} \mathbf{M})^{(4)}$ şi $(\mathbf{I} \dot{+} \mathbf{M})^{(8)}$.

* Între două mulţimi **C** şi **C'** există o relaţie de ordine tranzitivă şi antisimetrică ($\mathbf{C} < \mathbf{C}'$) dacă $\mathbf{C} \neq \mathbf{C}'$ şi dacă există un drum de la un vîrf a lui **C** la un vîrf a lui **C'**.

Scanned with OKEN Scanner

Însumînd matricea unitate **I** cu matricea **M** se obține $(\mathbf{I} \dot{+} \mathbf{M})$ și pe baza acesteia se calculează $(\mathbf{I} \dot{+} \mathbf{M})^{(2)}$:

$$(\mathbf{I} \dot{+} \mathbf{M})^{(2)} = (\mathbf{I} \dot{+} \mathbf{M}) \dot{\times} (\mathbf{I} \dot{+} \mathbf{M}) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

În continuare se calculează $(\mathbf{I} \dot{+} \mathbf{M})^{(4)}$:

$$(\mathbf{I} \dot{+} \mathbf{M})^{(4)} = (\mathbf{I} \dot{+} \mathbf{M})^{(2)} \dot{\times} (\mathbf{I} \dot{+} \mathbf{M})^{(2)} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

După calcularea matricii $(\mathbf{I} \dot{+} \mathbf{M})^{(8)} = (\mathbf{I} \dot{+} \mathbf{M})^{(4)} \dot{\times} (\mathbf{I} \dot{+} \mathbf{M})^{(4)}$ se constată că $(\mathbf{I} \dot{+} \mathbf{M})^{(4)} = (\mathbf{I} \dot{+} \mathbf{M})^{(8)}$, prin urmare este satisfăcută relația (3.8) și se poate trece la descompunerea grafului în clase de echivalență. În matricea $(\mathbf{I} \dot{+} \mathbf{M})^{(4)}$ sînt evidențiate trei clase de echivalență :

a) clasa $\mathbf{C}_1 = \{x_1\}$

b) clasa $\mathbf{C}_2 = \{x_2, x_3, x_4, x_5, x_6\}$

c) clasa $\mathbf{C}_3 = \{x_7\}$

În clasa de echivalență $\mathbf{C}_2$ există un singur drum hamiltonian : $(x_3, x_6, x_4, x_2, x_5)$. Ținînd seama de arcele care leagă cele trei clase de echivalență găsim drumul hamiltonian :

$$\mu_h = (x_1, x_3, x_6, x_4, x_2, x_5, x_7).$$

Capitolul IV

ANALIZA DRUMULUI CRITIC

4.1. Trecerea de la metodele tradiţionale de planificare a acţiunilor complexe la metodele ADC*

Cunoaşterea detaliată a „tehnologiei" acţiunilor complexe şi a eşalonării în timp a acestora a constituit o necesitate imperioasă în munca specialiştilor care organizează şi conduc astfel de acţiuni, în special odată cu creşterea din ultimele decenii a amploarei şi complexităţii activităţii economice.

Un instrument de mare utilitate în aceste probleme l-a constituit şi-l mai constituie încă graficul calendaristic tip Gantt, apărut la începutul secolului nostru.

Să considerăm proiectul unei investiţii importante a cărei durată şi mod de desfăşurare se doreşte a fi cunoscută la un nivel foarte general. Metoda graficului Gantt presupune divizarea acţiunii complexe pe care o prezintă investiţia, în părţi componente (lucrări) şi apoi eşalonarea acestora în timp, ţinînd seama de succesiuni tehnologice, resurse, termene impuse, conform figurii 4.1.

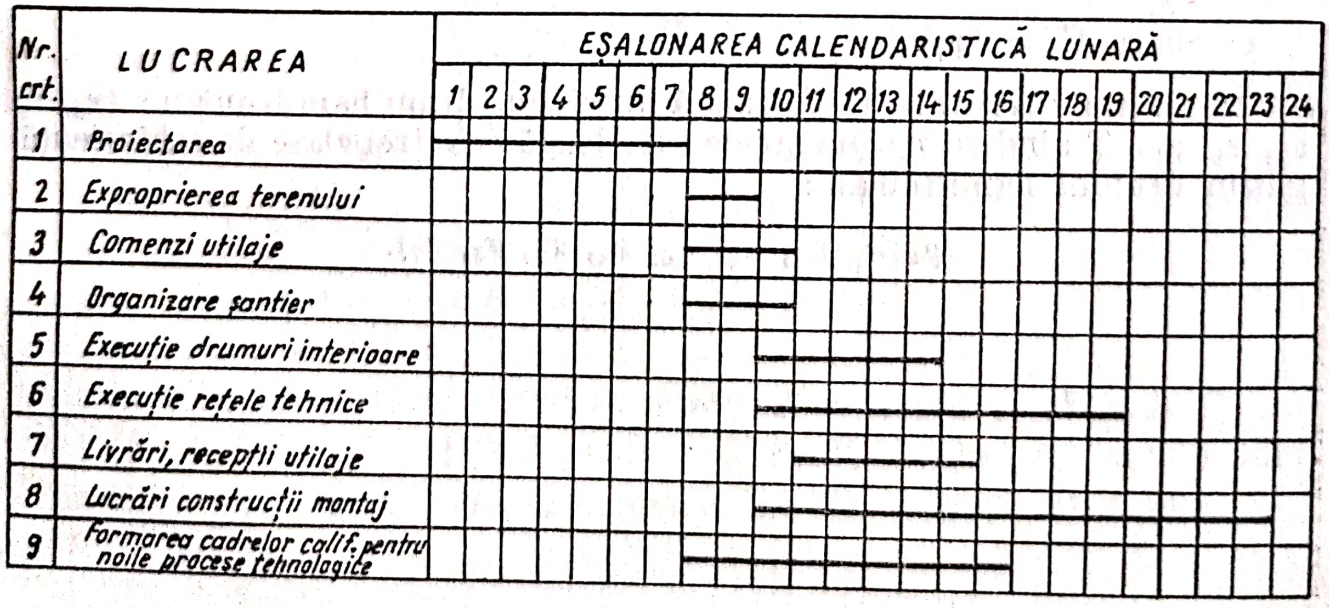


Fig. 4.1

* ADC — Analiza Drumului Critic.

Scanned with OKEN Scanner

Dacă este întocmit în urma unei analize temeinice, graficul Gantt oferă informaţii interesante şi extrem de sugestiv prezentate, privind desfăşurarea lucrărilor precum şi o serie de informaţii derivate privind eşalonarea resurselor necesare (forţe de muncă, materii prime şi materiale, utilaje, fonduri băneşti etc.). Aceste avantaje scad însă simţitor cînd, datorită fie amploarei acţiunii considerate, fie modului de detaliere dorit, numărul lucrărilor care compun graficul Gantt creşte mult ajungînd la cîteva sute sau mii. Urmarea creşterii numărului de lucrări componente este în primul rînd pierderea posibilităţii de corelare a acestora, fenomen pe care cu puţină imaginaţie îl putem ilustra şi referitor la exemplul redus din fig. 2.1. Pentru aceasta, însă, va trebui să arătăm deocamdată în principiu, cum se prezintă desfăşurarea aceleiaşi investiţii cu ajutorul unei reţele ADC.

Principiul acestei metode este divizarea acţiunii complexe în părţi componente la un nivel care să permită corelarea logică şi tehnologică a acestora, adică să facă posibilă stabilirea intercondiţionărilor între părţile componente.

Aceste părţi componente se numesc în metodele ADC activităţi *.

La definirea listei acţiunilor (prima operaţie din metodologia ADC), specialistul sau specialiştii care participă la această operaţie folosesc experienţa lor pentru a întreba în legătură cu fiecare activitate $\mathbf{A}_i$: ce alte activităţi condiţionează sau preced în mod necesar activitatea $\mathbf{A}_i$?

Ia naştere în felul acesta un tabel ca cel de mai jos, care conţine, pentru aceeaşi acţiune complexă, datorită jocului intercondiţionării, mai multe activităţi decît graficul Gantt.

Astfel, lucrarea „organizarea şantier" din graficul Gantt, cu durata de 3 luni, a fost divizată în două activităţi ($\mathbf{O.S_1}$ şi $\mathbf{O.S_2}$) cu durata de 1, respectiv 2 luni.

Durata totală a lucrării a fost deci menţinută, împărţirea ei fiind însă necesară pentru a pune în evidenţă faptul că activităţile $\mathbf{D_1}$, $\mathbf{R_1}$ şi $\mathbf{C_1}$ sînt condiţionate numai de prima parte a acestei lucrări.

Divizări în cîte două activităţi, pentru motive similare, s-au făcut şi pentru lucrările de drumuri interioare, reţele tehnice şi construcţii-montaje.

Elementele din tabelul 4.1 sînt suficiente pentru a construi reţeaua sau graful asociat acţiunii complexe pe care vrem să o reprezentăm prin metoda ADC, dacă se ţine seama de două convenţii:

— unei activităţi i se asociază un segment orientat numit arc;

— condiţionarea (succesiunea) a două activităţi se reprezintă prin succesiunea a două arce adiacente.

În figura 4.2 s-a figurat graful corespunzător tabelului 4.1.

La cele două convenţii de mai sus mai adăugăm şi pe următoarele:

— fiecare activitate (arc) are un moment de începere şi unul de terminare;

---

* Noţiunea de „activitate" din metodele ADC este perfect echivalentă cu cea de „lucrare" din metoda Gantt, dar pentru a sublinia diferenţierea metodologică, menţinem cele două denumiri în legătură cu metoda utilizată.

Scanned with OKEN Scanner

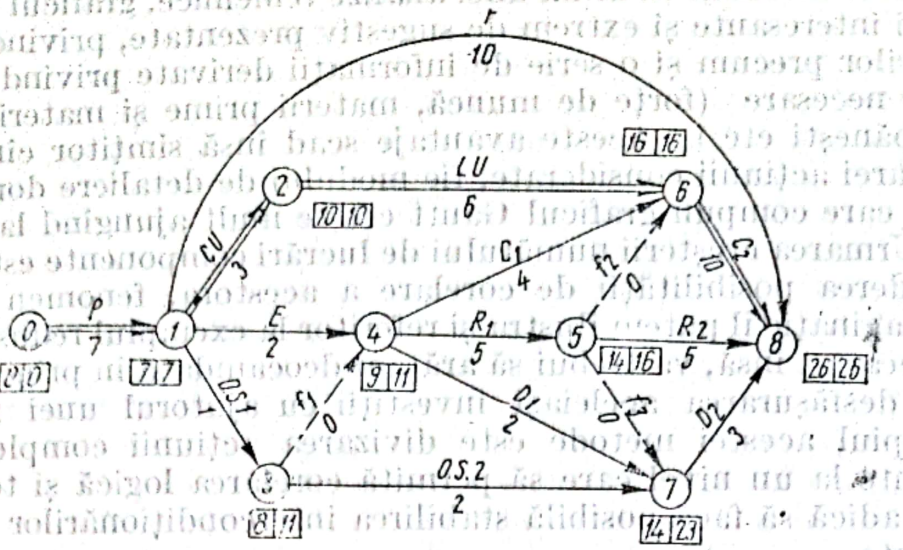


Fig. 4.2

— fiecărui arc i se asociază durata activității pe care o reprezintă, înscriindu-se această durată în grafic;

— nici o activitate nu poate începe înainte de terminarea tuturor activităților precedente și nu se poate termina după începerea activităților precedente și nu se poate termina după începerea activităților următoare.

Tabelul 4.

Nr. crt.	Denumirea activității și prescurtarea sa	Activitatea imediat precedentă	Durata (luni)
1.	Proiectarea (P)	—	7
2.	Exproprierea terenului (E)	P	2
3.	Comenzi utilaje (C.U.)	P	3
4.	Organizarea șantierului etapa 1 ($O.S_1$.)	P	1
5.	Formare cadre calificate (F)	P	10
6.	Execuție drumuri interioare — etapa 1 (D_1)	E; $O.S_1$	2
7.	Execuție rețele tehnice — etapa 1 (R_1)	E; $O.S_1$	5
8.	Livrări, recepții utilaje, (L.U.)	C.U.	6
9.	Lucrări construcții-montaje — etapa 1 (C_1)	E; $O.S_1$	4
10.	Organizare șantier etapa 2 ($O.S_2$)	$O.S_1$	2
11.	Execuție drumuri interioare — etapa 2 (D_2)	D_1; $O.S_2$; R_1	3
12.	Execuție rețele tehnice — etapa 2 (R_2)	R_1	5
13.	Lucrări construcții-montaje etapa 2 (C_2)	L.U.; C_1; R_1	10

Vom observa că în graf se poate lua în considerare diferite succesiuni de activități adiacente, începînd cu începerea primei dintre ele (în cazul nostru activitatea **P**) și terminînd cu terminarea ultimei (oricare din activitățile **F**, C_2, R_2, D_2 poate fi activitatea ultimă).

Scanned with OKEN Scanner

Există evident numeroase astfel de succesiuni cărora le corespund diferite durate totale. Astfel:

— Succesiunea (P.), (F): durata $7 + 10 = 17$ luni.

— Succesiunea (P.), (E.), (R_1), (R_2): durata $7 + 2 + 5 + 5 = 19$ luni.

— Succesiunea (P.), (O. S_1), (f_1)*, (D_1), (D_2), durată $7 + 1 + 0 + 2 + 3 = 13$ luni.

În secțiunea a 2-a a prezentului capitol vom prezenta metoda de determinare a succesiunii cu cea mai lungă durată, fără a fi necesară enumerarea tuturor succesiunilor din graf. Deocamdată vom arăta că în exemplul nostru această succesiune este (P), (C.U.), (L.U.), (C_2), avînd o durată corespunzătoare de $7 + 3 + 6 + 10 = 26$ luni.

Deosebit de importantă pentru sesizarea esenței metodei este înțelegerea faptului că cea mai „lungă" succesiune determină durata minimă, posibilă de execuție, a întregii acțiuni (în cazul nostru, realizarea investiției). Acest lucru este evident întrucît nici una dintre activități nu poate începe înainte de terminarea celor care le condiționează și în consecință nici ultima activitate a succesiunii de durată maximă.

Iată însă că apare o neconcordanță între durata totală obținută prin metoda Gantt și cea care rezultă din graful ADC: prima este de 22 luni, a doua (cea reală), de 26 luni. Acest lucru se datorește tocmai faptului pe care l-am arătat, că metoda Gantt nu pune în evidență cu suficientă finețe intercorelările dintre activități **.

Dacă aceasta ar fi unicul avantaj al metodei ADC, față de metoda tradițională a graficului calendaristic și încă ar fi deajuns pentru adoptarea metodei care indică durata reală a acțiunii. Evident, în realitate, cînd avem de-a face cu acțiuni ample și complexe de sute sau mii de activități, erorile pe care le-ar introduce folosirea metodei Gantt sînt considerabile.

După cum vom vedea însă, în cele ce urmează, mai sînt și alte avantaje ale metodelor ADC, tot atît de importante.

Anticipînd asupra acestei probleme la care vom reveni în detaliu pe parcursul prezentului capitol, vom arăta că principalele avantaje ale metodelor ADC sînt următoarele:

— obligă la o analiză amănunțită a structurii tehnologice și logice a acțiunii considerate;

— evidențiază explicit legăturile logice și tehnologice între operații (ceea ce metoda Gantt nu face);

— determină (după cum am arătat mai sus), durata reală a acțiunii complexe;

— evidențiază operațiile-cheie (așa-numitele „activități critice"), asupra cărora trebuie îndreptată în mod special atenția organizatorilor;

---

* În graful din figura 4.1. s-au introdus cîteva așa-numite „activități fictive" (f_1, f_2, f_3) reprezentate punctat, care au rolul de a preciza unele condiționări (de exemplu, fără activitatea fictivă f_1, nu este pusă în evidență condiționarea lui C_1, R_1 și P_1 de O.S_1).

** La nivelul exemplului nostru, desigur intercorelările dintre activități s-ar fi putut pune în evidență și printr-un grafic Gantt. Operația n-ar mai fi fost, însă, posibilă pentru cîteva sute sau mii de activități.

— evidenţiază „activităţile necritice" care permit eventuale amînări (aşa-numitele „rezerve") şi determină mărimea rezervelor, ceea ce face posibilă eventuale redistribuiri de resurse spre activităţile critice;

— odată constituit graful unei acţiuni complexe, dacă acesta este corect, graful îşi păstrează configuraţia chiar dacă unele condiţii obiective sau subiective concură la modificarea duratelor şi termenelor; acest fapt permite ca luarea în considerare a acestor modificări să se facă fără refacerea integrală a planificării (aşa cum este neapărat necesar în metoda GANTT de cîte ori apar modificări de durate şi termene);

— modelele ADC sînt după cum vom vedea, singurele metode care permit optimizarea utilizării în timp a resurselor (forţă de muncă, materiale, utilaje, fonduri băneşti etc.).

Toate avantajele de mai sus sînt posibile, în cazul acţiunilor complexe cu sute sau mii de activităţi, numai dacă se utilizează calculatoarele electronice care au, în majoritate, programe şi sisteme de programe pentru diverse probleme ADC.

În final, aceste avantaje „de lucru" se concretizează în importante avantaje economice (scurtări de durată în execuţia lucrărilor complexe, utilizarea raţională a resurselor, evitarea unor întreruperi etc.), care fac din Analiza drumului critic una din cele mai eficiente metode ale cercetării operaţionale.

4.2. Metoda CPM (Critical Path Method)

A. Reprezentarea activitate-arc

Considerînd că în secţiunea 4.1 am introdus noţiunile de activitate, arc, graf şi durată, să introducem în continuare alte noţiuni precum şi unele reguli şi precizări.

4.2.1. Activităţile sînt de trei feluri:

— propriu-zise: cele care consumă timp şi resurse;
— aşteptări: consumă numai timp;
— fictive: nu consumă nici timp nici resurse.

4.2.2. Evenimentele sînt momente caracteristice ale unei acţiuni complexe descrise printr-un graf ADC, reprezentînd stadii de realizare a activităţilor, respectiv terminarea uneia sau mai multor activităţi şi/sau începerea uneia sau mai multor activităţi. Evenimentele se reprezintă de regulă în grafe prin cerculeţe numite noduri care se numerotează de obicei secvenţial, conform regulii: „numărul nodului de început al unei activităţi trebuie să fie totdeauna mai mic decît numărul nodului de sfîrşit".

Scanned with OKEN Scanner

4.2.3. Orice graf ADC are un singur nod inițial și un singur nod terminal.

4.2.4. În grafele ADC nu sînt admise circuite, adică succesiuni de activități pentru care terminarea unei activități coincide cu începerea uneia din activitățile care o preced.

4.2.5. Orice activitate trebuie să aibă cel puțin o activitate precedentă și cel puțin una care îi succede, exceptînd bineînțeles activitățile care încep în nodul inițial și pe cele care se termină în nodul final.

4.2.6. Regula de figurare a activităților care încep în același nod:
— două activități care au comun nodul de începere nu pot avea comun și nodul de terminare (vezi fig. 4.3 : *a* — reprezentarea greșită; *b* și *c* — reprezentări corecte — (după caz);

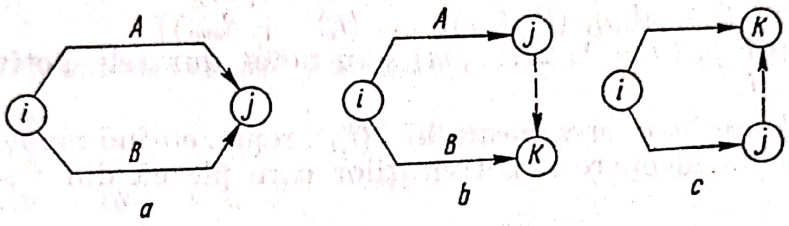


Fig. 4.3

4.2.7. Regula evitării introducerii unor dependențe nereale (neprevăzute în tabelul de condiționări) ca urmare a unor reprezentări eronate :
Fie tabelul de condiționări :

Tabelul 4.2

Activitatea	Activități imediat precedente
A	—
B	—
C	A
D	A,B

Reprezentarea conform figurii 4.4, *a* este eronată întrucît introduce dependența **C → B** neprevăzută în tabel. Reprezentarea corectă este cea din figura 4.4, *b*.

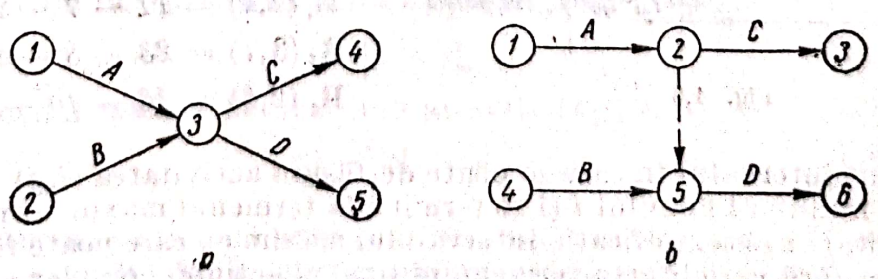


Fig. 4.4

4.2.8. Reguli privind aspectul geometric al rețelelor:
— activitățile nu se desenează în general la scară;
— se preferă liniile drepte orizontale cu sensul arcelor stînga-dreapta sau la cca 45° cu sens S.V. → N.E. sau N.V. → S.E.

4.2.9. Termenele evenimentelor

Fie nodul i care reprezintă evenimentul final al activităților (h_1, i), $(h_2, i), \ldots, (h_n, i)$ și în același timp evenimentul inițial al activităților $(i, j_1), (i, j_2), \ldots, (i, j_n)$ — vezi figura 4.5.

Termenul minim al evenimentului i (t_m^i), adică momentul cel mai devreme posibil de terminare a tuturor activităților care converg în i, se calculează astfel:

$t_m^o = 0$ (cu zero se numerotează nodul inițial)

$t_m^i = \max [(t_m^{h_1} + d_{h_1 i}), (t_m^{h_2}, t_{h_2 i}), \ldots, (t_m^{h_n} + d_{h_n i})]$

de unde prin $d_{h_g i}$, $(g = 1, 2, \ldots, m)$ s-au notat duratele activităților care converg în i.

Termenul maxim al evenimentului i (t_M^i), reprezentînd momentul cel mai tîrziu posibil de începere a activităților care pleacă din i se calculează astfel:

$t_M^f = t_m^f$

$t_M^i = t_m^f$ (f este numărul de ordine al nodului final)

$t_M^i = \min [(t_M^{j_1} - d_{ij_1}), (t_M^{j_2} - d_{ij_2}), \ldots, (t_M^{j_n} - d_{ij_n})]$

unde prin d_{ij_k}, $(k = 1, 2, \ldots, n)$, s-au notat duratele activităților care pleacă din i.

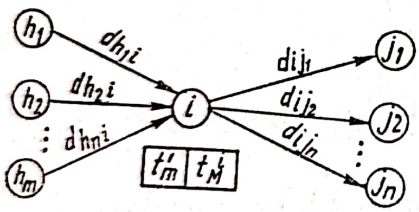


Fig. 4.5

Aplicînd formulele de mai sus în cazul grafului din figura 4.2 se obțin termenele minime și maxime înscrise în micile dreptunghiuri din dreptul fiecărui nod (vezi fig. 4.5).

4.2.10. Rezerva totală a unei activități $(i - j)$ se calculează cu expresia:

$$\mathbf{R}_t(i, j) = t_M^j - t_m^i - d_{ij}$$

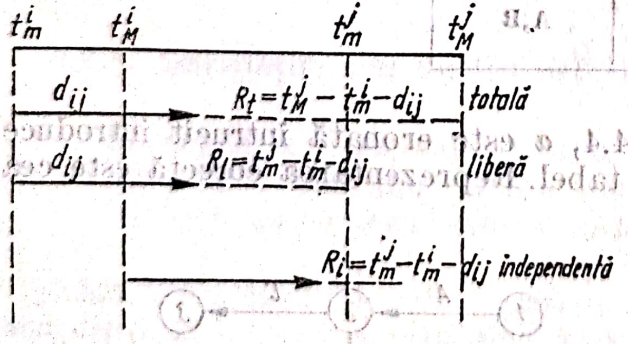


Fig. 4.6

Pentru exemplificare, să calculăm rezerva totală a cîtorva activități din graful reprezentat în figura 4.2.

$$\mathbf{R}_t(1,4) = 11 - 7 - 2 = 2$$

$$\mathbf{R}_t(3,7) = 23 - 8 - 2 = 13$$

$$\mathbf{R}_t(2,6) = 16 - 10 - 6 = 0$$

Întrucît intervalul în care se poate desfășura activitatea (i, j) începe la termenul minim al nodului i și se termină la termenul maxim al nodului j, rezerva totală are semnificația intervalului maxim cu care poate fi amînată activitatea fără periclitarea termenului final al acțiunii complexe.

Scanned with OKEN Scanner

4.2.11. Rezerva liberă a unei activităţi $i - j$ se calculează cu expresia :

$$R_1(i, j) = t_m^j - t_m^i - d_{ij}$$

Vom avea (vezi figura 4.2) :

$$R_1\ (1,4) = 9 - 7 - 2 = 0$$
$$R_1\ (3,7) = 14 - 8 - 2 = 4$$
$$R_1\ (2,6) = 16 - 10 - 6 = 0$$

Rezerva liberă are semnificaţia intervalului cu care poate fi amînată o activitate fără a consuma din rezerva activităţilor care îi succed.

4.2.12. Rezerva independentă a unei activităţi $(i - j)$ are expresia :

$$R_I(i, j) = t_m^j - t_M^i - d_{ij}$$

Pentru graful din figura 5.2 vom avea :

$$R_I(1,4) = 9 - 7 - 2 = 0$$
$$R_I(3,7) = 14 - 11 - 2 = 1$$
$$R_I(2,6) = 16 - 10 - 6 = 0$$

Această rezervă ne arată intervalul în care poate fi „plasată" o activitate fără a consuma rezervele activităţilor care îi succed şi care îi preced.

4.2.13. Termenele de începere şi de terminare ale activităţilor

O activitate $i - j$ se poate desfăşura între termenul minim al evenimentului i şi termenul maxim al evenimentului j.

Dacă activitatea $(i - j)$ nu dispune de rezervă totală începutul ei va coincide obligatoriu cu termenul minim al evenimentului i iar sfîrşitul ei cu termenul maxim al evenimentului j.

Dacă însă există o rezervă totală această coincidenţă nu se mai realizează, activitatea $(i - j)$ putîndu-se „glisa" între t_m^i şi t_M^j (fig. 4.6).

Avînd în vedere această situaţie vom defini :

— Termenul minim de începere al activităţii $(i - j)$:

$$t_m^{ij} = t_m^i$$

— Termenul minim de terminare al activităţii $(i - j)$:

$$t_{mt}^{ij} = t_m^i + d_{ij}$$

— Termenul maxim de terminare al activităţii $(i - j)$:

$$t_{Mt}^{ij} = t_M^j$$

— Termenul maxim de începere al activităţii $(i - j)$:

$$t_{Mi}^{ij} = t_M^j - d_{ij}$$

4.2.14. Drumul critic al unui graf este succesiunea activităţilor dintre nodul iniţial şi nodul final, care au rezerva totală nulă.

Scanned with OKEN Scanner

Drumul critic reprezintă în mod evident succesiunea cu durată maximă din graf (în cazul figurii 4.2, activităţile de pe drumul critic au fost desenate dublat).

Prin tehnica calculului drumului critic se găseşte deci, direct, durata minimă posibilă de realizare a acţiunii complexe.

B. Reprezentarea activitate-nod

O variantă a metodei CPM utilizează reprezentarea activităţilor prin noduri, iar a condiţionărilor prin arce.

4.2.15. Fiecare nod are şase compartimente (fig. 4.7) în care se înscriu (pentru o activitate i) :

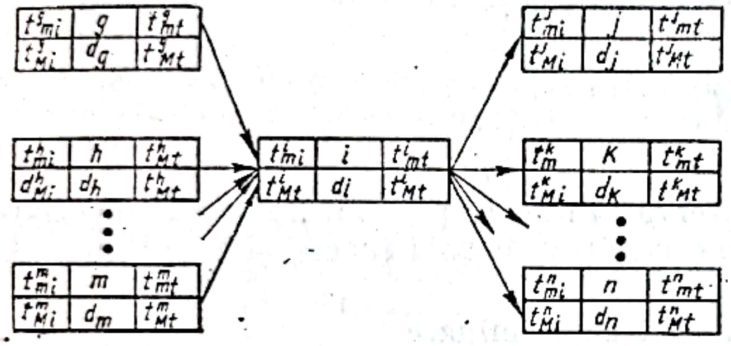


Fig. 4.7

— numărul (simbolul) activităţii : i ;

— durata activităţii : d_i ;

— termenul minim de începere al activităţii considerată programată cel mai devreme posibil în intervalul aferent t^i_{mi} ;

— termenul minim de terminare (în condiţiile precizate mai sus) t^i_{mt} ;

— termenul maxim de începere cînd programarea se face cel mai tîrziu posibil în intervalul aferent : t^i_{Mi} ;

— termenul maxim de terminare : t^i_{Mt} ;

4.2.16. Din condiţia ca o activitate i să nu înceapă pînă la terminarea tuturor celor care o condiţionează, rezultă :

$$t^i_{mi} = \max\,[t^o_{mt},\ t^h_{mt}, \ldots,\ t^m_{mt}]$$

$$t^i_{mt} = t^i_{mi} + d_i,$$

iar din condiţia ca o activitate să nu se termine după începerea celor pe care le condiţionează :

$$t^i_{Mi} = \min\,[t^j_{Mi},\ t^k_{Mi}, \ldots,\ t^n_{Mi}]$$

$$t^i_{Mi} = t^i_{Mt} - d_i.$$

4.2.17. Rezerva totală se va determina astfel :

$$R^i_t = t^i_{Mt} - t^i_{mi} - d_i$$

dar

$$t^i_{Mt} - d_i = t^i_{Mi},$$

deci :

$$R^i_t = t^i_{Mi} - t^i_{mi}.$$

Scanned with OKEN Scanner

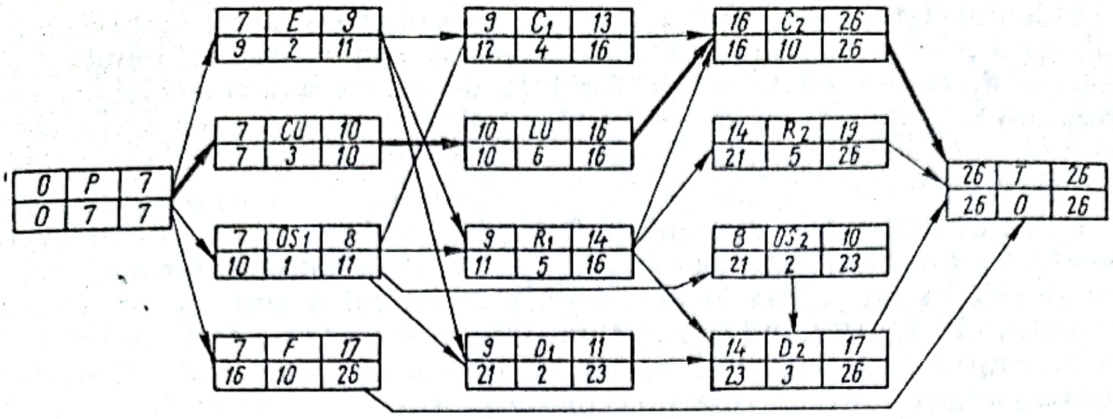


Fig. 4.8

Aplicînd formulele de mai sus se obţin termenele, rezervele şi drumul critic al grafelor reprezentate pe noduri.

În figura 4.8 s-a construit şi calculat în reprezentarea pe noduri graful CPM din exemplul nostru de mai sus.

Din necesitatea ca graful să aibă un singur nod pe început şi un singur nod de sfîrşit s-a introdus nodul fictiv de terminare T.

Se observă uşor identitatea de fond a grafelor 2.2 şi 4.8.

C. Calculul termenelor, rezervelor şi drumului critic prin tabel

4.2.18. Toate elementele semnificative pe care le furnizează metoda CPM cu reprezentarea pe arce sau noduri se pot obţine şi cu ajutorul întocmirii unui tabel conform celui de mai jos (tabelul 4.3).

Tabelul 4.3.

Nr. crt.	Denumire activ.	Activ imediat precedente	Durata	Termene				Rezerva totală	Obs.
				minime		maxime			
				Începere t^{mi}	Terminare t^{mt}	Începere t^{Mi}	Terminare t^{Mt}		
0	1	2	3	4	5	6	7	8	9
1	P	—	7	0	7	0	7	0	D.C.
2	CU	P	3	7	10	7	10	0	D.C.
3	OS1	P	1	7	8	10	11	3	
4	E	P	2	7	9	9	11	2	
5	F	P	10	7	17	16	26	9	
6	LU	CU	6	10	16	10	16	0	D.C.
7	OS2	OS1	2	8	10	21	23	13	
8	R_1	E ; OS_1	5	9	14	11	16	2	
9	C_1	E ; OS_1	4	9	13	12	16	3	
10	D_1	E ; OS_1	2	9	11	21	23	12	
11	R_2	R_1	5	14	19	21	26	7	
12	C_2	LU ; C_1 ; R_1	10	16	26	16	26	0	D.C.
13	D_2	OS_2 ; D_1 ; R_1	3	14	17	23	26	9	
14	T	F ; R_2 ; C_2 ; D_2	0	26	26	26	26	0	D.C.

Scanned with OKEN Scanner

Calculul termenelor minime se face pornind de la $t^P_{mi} = 0$ (rîndul 1, coloana 4), după care elementele din coloana 5 se obţin cu formula $t_{mi} = t_{mi} + d$, ţinînd seama şi de formula de calcul a termenelor minime prezentată la 4.2.16 (cînd completăm coloana 4 pentru o activitate i) luăm cea mai mare din valorile înscrise în coloana 5 în dreptul activităţilor care preced pe i).

Calculul termenelor maxime se face plecînd de la $t^T_{Mi} = 26$ (rîndul 14, coloana 7), după care elementele din coloana 6 se obţin cu formula $t_{Mi} = t_{Mi} - d$, ţinînd seama şi de formula de calcul a termenelor maxime prezentate la 4.2.16 (cînd completăm coloana 7 pentru o activitate i luăm cea mai mică dintre valorile înscrise în coloana 6 în dreptul activităţilor care au ca activitate imediat precedentă pe i).

În coloana 8 se înscriu rezervele totale, iar în coloana 9 se notează activităţile de pe drumul critic.

4.3. Grafice ADC* integrate şi condensate

În practica conducerii acţiunilor economice complexe prin metodele ADC, nivelul de detaliere în activităţi a acţiunilor în ansamblul lor, depinde de scopul urmărit (coordonare de ansamblu sau conducere de detaliu), de timpul avut la dispoziţie pentru elaborarea grafelor, de specialiştii disponibili etc.

Dacă graful sumar care se întocmeşte pentru orientarea generală a echipei de conducere a acţiunii (să-l numim graf director) este totdeauna necesar, cînd se face trecerea la conducerea de amănunt, sînt tot atît de necesare grafele detaliate.

În general grafele detaliate se fac pe părţi din acţiunea complexă, numite obiective sau obiecte. Astfel, dacă ne referim la exemplul de mai sus, graful din figura 4.2 care este un graf director sumar, poate fi divizat în grafe pe obiect, după cum urmează:

— graful proiectării;
— graful organizării de şantier;
— unul sau mai multe grafe pentru lucrări de drumuri;
— mai multe grafe pentru lucrări de reţele (apă, canal, abur, electrice etc.);
— mai multe grafe pentru lucrări de construcţii-montaj (cîte unul pentru fiecare hală sau clădire) etc.

Grafele pe obiect au individualitatea lor şi se tratează ca entităţi de programare distincte, în acelaşi timp însă trebuie gîndită coordonarea lor în cadrul acţiunii complexe din care fac parte. În acest scop, după întocmirea separată a grafelor pe obiect apare necesitatea asamblării lor într-un tot care constituie graful integrat.

* În prezentul capitol am utilizat, cu acelaşi înţeles, termenul *grafic-reţea*, alături de termenele *graf* şi *reţea*.

Scanned with OKEN Scanner

Construirea şi calculul grafelor integrate au cîteva particularităţi: Fie **A, B,..., P** mai multe grafe ADC pe obiect, reprezentate schematic în figura 4.9.

Fiecare din grafe are o numărătoare proprie a evenimentelor.

Între activităţile diferitelor grafe există condiţionări logice şi tehnologice reprezentate prin activităţi fictive.

$$(a_2 \to b_2,\ b_3 \to p_2,\ p_3 \to b_4,\ p_4 \to a_3),$$

Grafele **A, B,..., P,** vor putea fi considerate reunite într-un graf integrat atunci cînd va fi introdus un nod iniţial fictiv *I* care va fi legat la toate nodurile iniţiale ale grafelor **A, B,..., P,** prin activităţi fictive şi un alt nod final fictiv **F,** de care se vor lega toate nodurile finale ale grafelor specificate prin activităţi fictive.

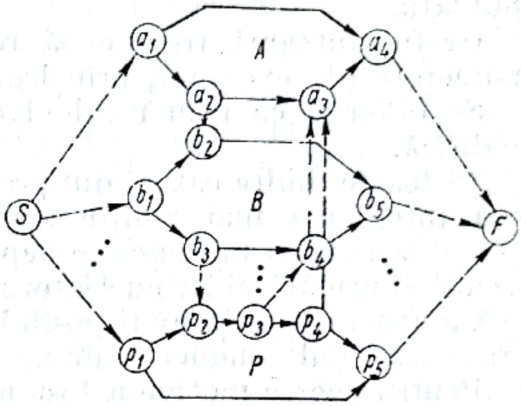


Fig. 4.9

După cum am arătat, la nivelul conducerii operative interesează construirea şi urmărirea grafelor pe obiecte, deci determinarea drumului critic pentru fiecare graf în parte. La nivelul coordonării întregii acţiuni, va fi necesară cunoaşterea drumului critic pentru graful integrat. Acesta, de regulă, diferă la fiecare din drumurile critice ale grafelor componente.

Pentru a ilustra modul cum se trece de la calculul drumului critic al grafelor componente, la calculul drumului critic al grafului integrat să urmărim exemplul din figura 4.10.

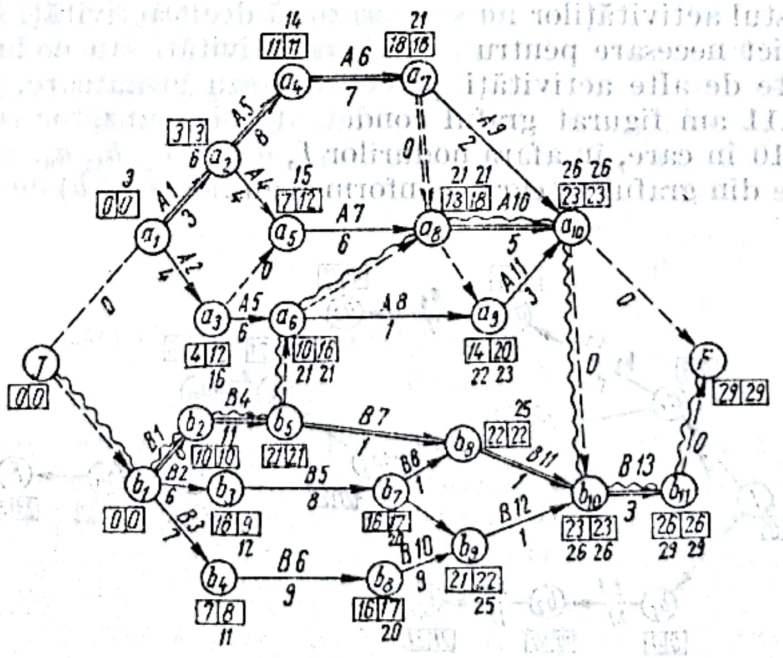


Fig. 4.10

Graful **A**, cu activităţile $A_1 - A_{11}$ are drumul critic $A_1 - A_3 - A_6 - A_{10}$, iar graful **B** cu activităţile $B_1 - B_{13}$ are drumul critic $B_1 - B_4 - B_7 - B_{11} - B_{13}$.

Prin introducerea legăturilor între activităţile $B_4 \rightarrow (A_8, A_{10})$ şi $(A_9, A_{10}, A_{11}) \rightarrow B_{13}$, în urma recalculării termenelor se găseşte drumul critic al grafului integrat : $I - B_1 - B_2 - A_{10} - B_{13} - F$.

În figura 4.10 drumurile critice ale graficelor **A** şi **B** s-au figurat prin dublarea arcelor, iar drumul critic al grafului integrat, printr-o linie ondulată.

Graful integrat trebuie să respecte toate condiţiile de construcţie enumerate (de exemplu, prin legăturile intergrafe să nu apară circuite).

Se observă că prin recalculare, unele termene ale evenimentelor se modifică.

În foarte multe cazuri din practică, numărul activităţilor care rezultă prin integrarea mai multor grafe pe obiect, este considerabil, putînd ajunge la zeci de mii ceea ce depăşeşte de multe ori posibilitatea de a le calcula şi urmări, chiar cu ajutorul calculatoarelor puternice.

Cu atît mai puţin ar fi posibil cuprinderea sintetică a unui asemenea graf, la nivelul conducerii întregii acţiuni.

Pentru aceste motive a fost necesară găsirea unui mijloc de a reduce graful integrat, păstrîndu-i în acelaşi timp principalele caracteristici. Această operaţie poartă numele de condensare şi se face după următoarele reguli :

a) graful condensat va conţine în mod obligatoriu nodurile de început şi sfîrşit ale grafului integrat şi ale fiecăruia din grafele pe obiect componente ;

b) el va cuprinde de asemenea toate activităţile şi nodurile de pe drumul critic ;

c) în graful condensat se pot reprezenta orice activităţi considerate deosebit de importante şi care trebuie explicitate ;

d) din restul activităţilor nu se reprezintă decît activităţi sau grupe de activităţi strict necesare pentru a nu lăsa activităţi sau noduri „în aer", adică nelegate de alte activităţi precedente sau următoare.

În fig. 4.11 am figurat graful condensat corespunzător celui integrat din figura 4.10 în care, în afara nodurilor I, a_1, b_1, b_5, b_6, a_6, a_8, a_{10}, b_{10}, b_{11} şi F, păstrate din graful integrat conform regulilor *a*) şi *b*) de mai sus, am

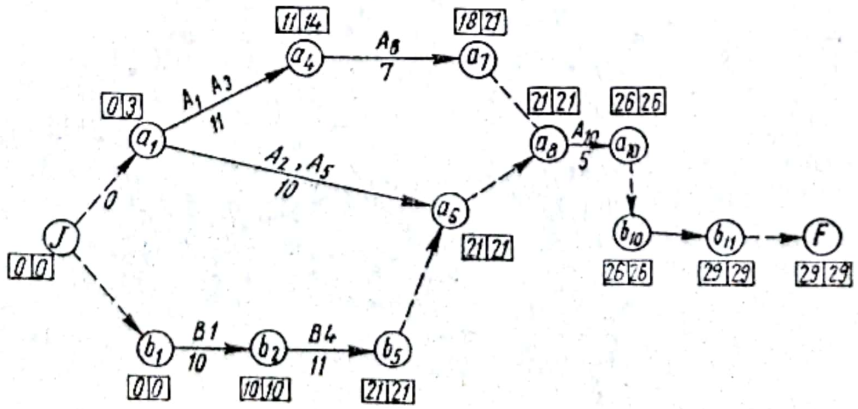


Fig. 4.11

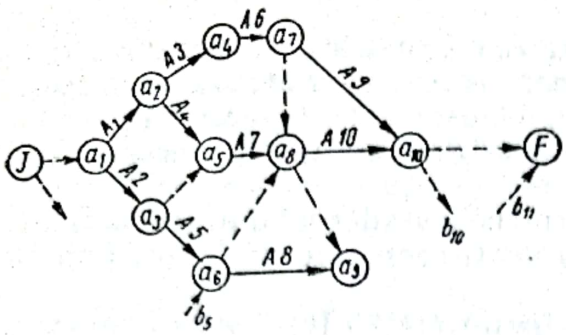


Fig. 4.12

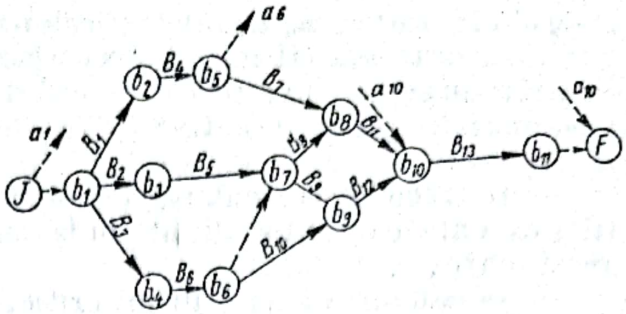


Fig. 4.13

considerat că activitatea $\mathbf{A}_6$ trebuie reprezentată avînd o importanţă deosebită. Ca urmare a acestui fapt şi a aplicării regulilor *c*) şi *d*), au rezultat în graful integrat şi nodurile a_4 şi a_7, precum şi activităţile $(\mathbf{A}_1, \mathbf{A}_3)$, $(\mathbf{A}_2, \mathbf{A}_5)$ şi activitatea fictivă dintre nodul a_7 şi a_8.

Prin condensare au rămas 14 activităţi din totalul de 35. În cazul grafelor mari şi foarte mari, condensarea este însă de regulă mult mai intensă, numărul activităţilor păstrate putînd reprezenta 10—20% din totalul celor din graful integrat, ceea ce reprezintă, evident, o simplificare considerabilă.

Legătura dintre diferitele grafe care alcătuiesc graful integrat se poate evidenţia cu ajutorul aşa-numitelor noduri de conexiune. Acestea au în primul rînd rolul de a permite desemnarea grafelor cu foarte multe activităţi şi care nu încap pe o singură foaie de hîrtie, ci pe mai multe foi. De exemplu graful integrat din figura 4.10 poate fi împărţit în cele două grafe componente şi deci desenat pe două foi distincte în felul următor (fig. 4.12 şi 4.13).

După cum se poate observa din unele noduri pleacă şi, spre alte noduri intră, arce rămase „în aer", care au exclusiv rolul de a indica legătura cu alte grafe. Aceste noduri (I, a_6, a_{10} şi F — pentru primul graf, I, b_3, b_{10} şi F — pentru al doilea graf) sînt noduri de conexiune. Cele două grafe se pot calcula independent, ţinînd seama atît la calculul termenelor minime cît şi la cel al termenelor maxime de influenţa termenelor din celălalt graf, cu ajutorul arcelor care intră în nodurile de conexiune.

4.4. Actualizarea graficelor ADC

În practica realizării acţiunilor complexe sînt numeroase cazurile cînd estimările iniţiale de durată ale activităţilor nu pot fi respectate. Apare astfel necesitatea ca, periodic, să se examineze modul cum se realizează termenele calculate, în scopul punerii în evidenţă a eventualelor întîrzieri şi a luării măsurilor de recuperare a acestora.

Această activitate poartă numele de actualizare a graficelor reţea.

Scanned with OKEN Scanner

Tehnica de actualizare a graficelor-reţea poate fi descrisă succint astfel :

— la data actualizării se examinează care activităţi sînt terminate, care sînt în curs de execuţie şi care sînt încă neatacate. Cu această ocazie se reestimează duratele activităţilor în curs de execuţie precum şi cele neatacate ;

— se trece la recalcularea noilor termene considerînd duratele activităţilor executate ca avînd durata nulă, iar pentru restul activităţilor, duratele reestimate ;

— se calculează noul drum critic, fie durata sa $\mathbf{D}_{c_t}$. Dacă momentul în care se face actualizarea este $\mathbf{T}_t$, noua estimare a duratei acţiunii va fi $\mathbf{D}_t = \mathbf{T}_t + \mathbf{D}_{c_t}$. Dacă această nouă durată este egală sau mai mică decît cea iniţială (**D**), nu sînt necesare măsuri speciale, deoarece lucrarea se va încadra în termenul stabilit. Dacă, dimpotrivă, $\mathbf{D}_t > \mathbf{D}$, se vor lua măsuri de scurtare a lui $\mathbf{D}_{c_t}$, prin suplimentări sau redistribuiri de resurse.

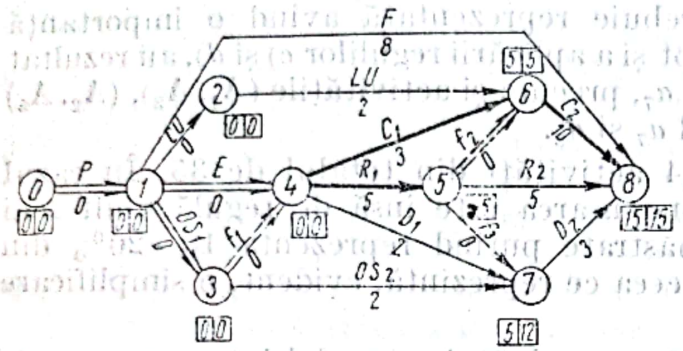


Fig. 4.14

Exemplu. Considerăm că la momentul $\mathbf{T}_t = 12$ se actualizează graful din figura 4.1, din reestimarea duratelor rezultînd datele înscrise în figura 4.14.

Recalculînd termenele, rezultă două drumuri critice : **P** ; **E** ; $\mathbf{R}_1$; $\mathbf{C}_2$ (lungimea $\mathbf{D}_1 = 0 + 0 + 3 + 10 = 13$) şi **P** ; **OS1** ; f_1 ; $\mathbf{R}_1$; $\mathbf{C}_2$ (lungimea $\mathbf{D}_1 = 0 + 0 + 0 + 3 + 10 = 13$). Vom avea $\mathbf{D}_t = 12 + 13 = 25$, deci ne încadrăm în termenul stabilit iniţial (26 unităţi de timp, conform fig. 4.2).

Tehnica de reactualizare descrisă mai sus este, evident, valabilă, cînd la momentul $\mathbf{T}_t$ al reactualizării, succesiunile şi condiţionările între activităţile neexecutate nu se modifică. Cînd apar astfel de modificări, odată cu reevaluarea duratelor se stabilesc noile condiţionări, operînd modificările respective în graful refăcut. Întrucît însă astfel de situaţii sînt relativ rare, procedeul de actualizare a graficelor reţea rămîne foarte operativ, incomparabil mai simplu decît reactualizarea graficelor tip Gantt care necesită de fiecare dată refacerea integrală a graficului.

4.5. Optimizări cost-durată

Estimarea duratelor acţiunilor complexe prin metodele ADC prezentate în paragrafele precedente reprezintă o problemă deosebit de importantă din punct de vedere economic (problema ADC/TIMP), însă nu singurul aspect care poate fi urmărit cu ajutorul acestor metode.

O altă problemă în care pot fi utilizate instrumentele ADC sînt cele de analiză a costului execuţiei acţiunilor complexe în funcţie de durata de execuţie.

Scanned with OKEN Scanner

În general se consideră că durata de execuţie a unei acţiuni complexe poate varia între o durată minimă posibilă T_{min} şi una maximă T_{max}. Evident, durata lucrării are numeroase implicaţii asupra costului, drept care prezintă un deosebit interes determinarea programului de lucru căreia îi corespunde un cost minim. În cele ce urmează vom prezenta succint această problemă.

4.5.1. Costurile unei activităţi

Vom considera că o activitate oarecare i a unei acţiuni complexe se poate desfăşura cu o durată d_i care, din punct de vedere tehnologic se situează între o limită inferioară d_1 şi una superioară d_2 ($d_1 \leqslant d_i \leqslant d_2$).

De asemenea vom mai considera că mărimea costului activităţii (c_i) este funcţie de această durată: $c_i = f(d_i)$. Dacă se consideră d_2 drept durată normală la execuţie a activităţii, este evident că o scurtare a duratei va conduce la creşterea costului. Dependenţa funcţională între c_i şi d_i poate fi foarte complexă (pătratică, neliniară, concavă sau chiar discretă), însă ea poate fi aproximată cu o funcţie liniară

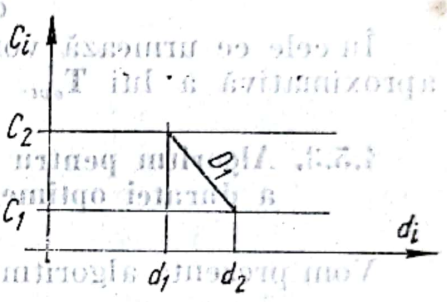


Fig. 4 .15

$$c_i = b_i d_i + a_i$$

care reprezintă o dreaptă (fig. 4.15)

În figura 4.15. d_1 este durata minimă a activităţii, d_2 durata maximă (normală), c_1 costul asociat duratei normale, iar c_2 costul duratei minime.

Din cunoştinţele elementare de geometrie analitică vom putea scrie ecuaţia dreptei (D):

$$\frac{d_i - d_1}{d_2 - d_1} = \frac{c_2 - c_i}{c_2 - c_1}$$

de unde va rezulta relaţia:

$$c_i = \frac{d_2 c_2 - d_1 c_1}{d_2 - d_1} - \frac{c_2 - c_1}{d_2 - d_1} d_i$$

pe care o vom putea pune sub forma: $c_i = a_i + b_i d_i$,
pentru a pune în evidenţă descreşterea costului odată cu creşterea duratei (ceea ce este echivalent cu costul unitar al urgentării lucrării).

4.5.2. Costul total al acţiunilor complexe

În general costul total al unei acţiuni complexe are o structură identică cu acela al unei investiţii fiind format din:

— costuri directe (C_D) — legate nemijlocit de realizarea activităţilor (costul resurselor, manoperei etc.);

Scanned with OKEN Scanner

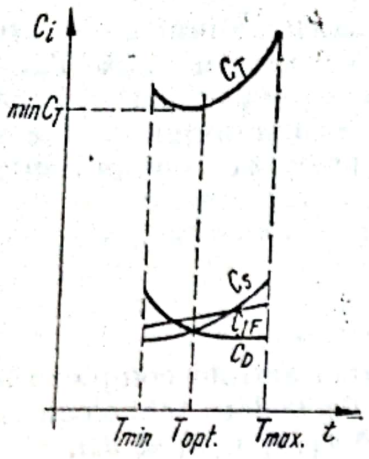


Fig. 4.16

— costuri indirecte (C_i) — cheltuieli generale, salariile personalului tehnic-administrativ, alte cheltuieli de regie;

— costul imobilizării fondurilor (C_{IF}) — pe perioade cînd investiția nu intră în funcțiune.

Toate aceste costuri sînt, evident, funcție de durata de execuție a investiției. În figura 4.16 se reprezintă alura generală a funcțiilor $C_D = f_1(t)$, $C_i = f_2(t)$ și $C_{IF} = f_3(t)$ în care T reprezintă durata totală a investiției.

Curba C_T reprezintă alura curbei funcției-sumă a celor trei funcții luate în considerare iar în grafic se poate citi timpul optim de execuție al acțiunii complexe (T_{opt}), corespunzător costului total minim (min C_T).

În cele ce urmează vom prezenta un algoritm pentru determinarea aproximativă a lui T_{opt}.

4.5.3. Algoritm pentru determinarea aproximativă a duratei optime de execuție a unei acțiuni complexe

Vom prezenta algoritmul cu ajutorul unui exemplu.

● *Etapa 1*

Fie acțiunea complexă descrisă în graful din figura 4.2, la care estimăm, pentru fiecare activitate, cîte o durată minimă și maximă și cîte un cost minim și maxim, conform tabelului 4.4.

Observăm că duratele maxime (normale) sînt cu mici modificări chiar cele luate în considerare în graful din figura 4.2.

Tabelul 4.4.

Nr. crt.	Denumirea activ.	Durata min. (d_1)	Durata max. (d_2)	Cost (milioane lei) min (C_1)	Cost (milioane lei) max (C_2)	Cost unitar al urgentării $C_1 = \frac{C_2 - C_1}{d_2 - d_1}$
1	P	6	7	1	2	1
2	E	2	2	1	1	0
3	CU	3	3	1	1	0
4	OS1	1	1	1	1	0
5	F	8	10	2	4	1
6	D_1	2	2	5	5	0
7	R_1	4	6	9	12	1,5
8	LU	4	6	20	25	2,5
9	C1	3	5	20	23	1,5
10	OS2	2	2	1	1	0
11	D_2	3	3	6	6	0
12	R_2	4	6	8	12	2
13	C_2	8	10	35	45	5
	Total:			110	138	

Scanned with OKEN Scanner

Ţinînd seama de aceste modificări, reconstruim graful cu noile termene (fig. 4.17).

Durata totală a rămas identică cu cea din figura 4.2 (26 luni) iar costul corespunzător este de 1,10 milioane lei (tabelul 4.4).

Vom numi această soluţie a problemei cost-durată, *programul* S_1.

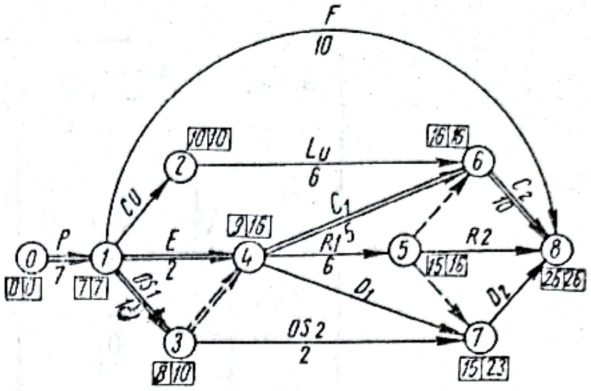


Fig. 4.17

● *Etapa 2*

Programul S_2 va consta în adoptarea duratelor minime şi a costurilor corespunzătoare (maxime), pentru fiecare activitate.

Aşadar, conform programului S_2 se estimează o durată totală de 21 luni (deci o scurtare de 5 luni faţă de programul S_1), ceea ce duce la o scumpire a costului de la 1,10 milioane la 1,38 milioane (figura 4.18) şi tabelul 4.4.

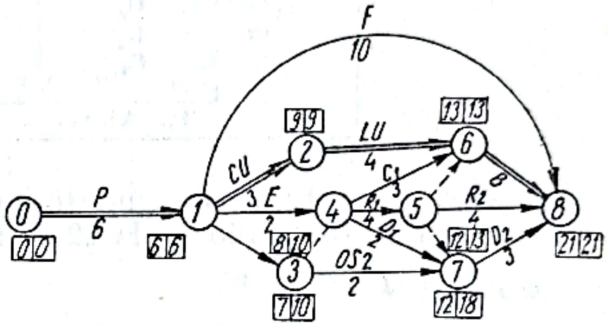


Fig. 4.1 8

● *Etapa 3*

Se trece la adoptarea unui *program* S_3 care menţine durata redusă (22 luni), dar încearcă ieftinirea costului prin prelungirea execuţiei duratelor necritice (fig. 4.19 şi tabelul 4.5).

Constatăm că au mai devenit critice şi activităţile **E**, $\mathbf{R}_1$ şi $\mathbf{C}_1$.

Prin prelungirea, în limita rezervelor, a duratelor activităţilor **F**, $\mathbf{R}_1$, $\mathbf{C}_1$ şi $\mathbf{R}_2$, costurile corespunzătoare se diminuează conform formulei :

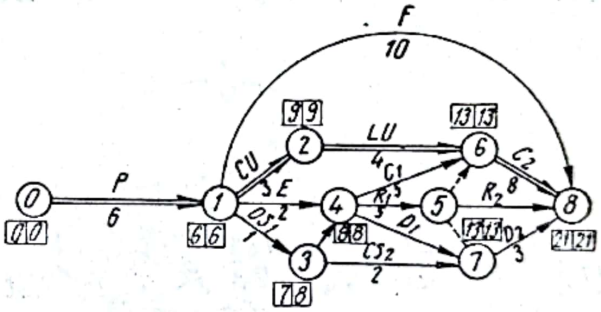


Fig. 4.19

$$c_i = a_i + b_i d_i$$

în care a_i este costul corespunzător duratei acestor activităţi în programul S_2, iar b_i este scurtarea lor (între estimările de durată din programul S_2 şi programul S_3).

$$c_5 = 4 - 1 \cdot (10 - 8) = 2$$

$$c_7 = 12 - 1{,}5 \cdot (5 - 4) = 10{,}5$$

$$c_9 = 23 - 1{,}5 \cdot (5 - 3) = 20$$

$$c_{12} = 12 - 2 \cdot (6 - 4) = 8.$$

Scanned with OKEN Scanner

Tabelul 4.5.

Nr. crt.	Denumirea activității	Durata d (luni)	Cost c (milioane)
1	P	6	2
2	E	2	1
3	CU	3	11
4	OS1	1	1
5	F	10	2
6	D_1	2	5
7	R_1	5	10,5
8	LU	4	25
9	C_1	5	20
10	OS_2	2	1
11	D_2	3	6
12	R_2	6	8
13	C_2	8	45
TOTAL :			127,5

Așadar, programul S_3, menținînd durata redusă cu 5 luni, reduce consumul de la 138 milioane la 127,5 milioane.

● *Etapa 4*

După cum am arătat, costul total al unei acțiuni complexe (de tipul investițiilor) este compus din costuri directe, indirecte și din costul imobilizărilor. Întrucît algoritmul urmărește minimizarea costurilor totale, este necesar să cunoaștem în primul rînd alura curbei cheltuielilor directe. Avem deja două puncte ale curbei, mai este necesar cel puțin unul. Pentru aceasta elaborăm un *program* S_4 cu durata între cea maximă (26 luni) și cea minimă (21 luni) prin analiza unor activități de pe drumul critic și prelungirea lor. Să considerăm programul S_4 cel din figura 4.20.

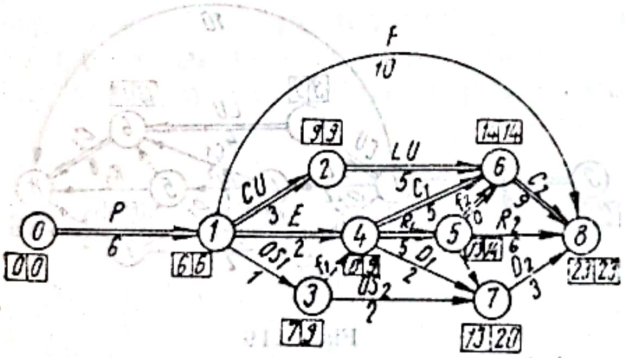


Fig. 4.20

● *Etapa 5*

Se optimizează programul S_4 prin prelungirea acelor activități necritice care mai pot fi prelungite (în cazul nostru doar R_1 se poate prelungi de la 5 la 6 luni). Obținem astfel *programul* S_5 (fig. 4.21).

Costul aferent programului S_5 se determină în tabelul 4.6, în care costurile c_i s-au calculat în felul următor :

$$c_7 = 10{,}5 - 1{,}5\ (6 - 5) = 9$$

$$c_8 = 25 - 2{,}5\ (5 - 4) = 22{,}5$$

$$c_{13} = 45 - 5 \cdot (9 - 8) = 40$$

Scanned with OKEN Scanner

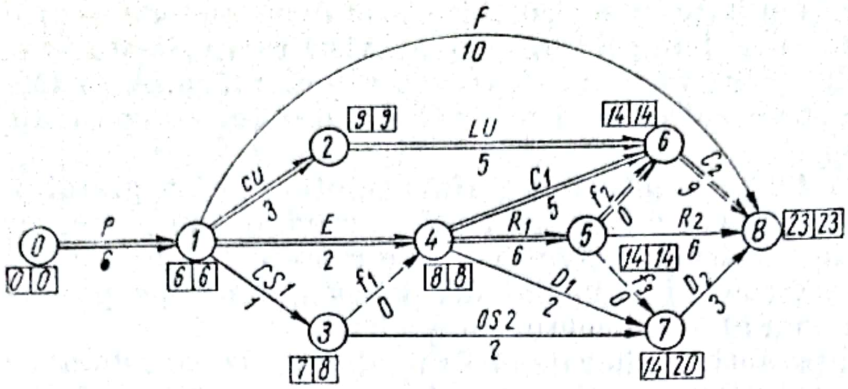


Fig. 4.21

Tabelul 4.6

Nr. crt.	Denumirea activității	Durata d_i (luni)	Cost c_i (milioane)
1	P	6	2
2	E	2	1
3	CU	3	1
4	OS1	1	1
5	F	10	2
6	D_1	2	5
7	R_1	6	9
8	LU	5	22,5
9	C1	5	20
10	OS_2	2	1
11	D_2	3	6
12	R_1	6	8
13	C_2	9	40
TOTAL:			118,5

Deci programul S_5 are o durată de 23 luni și un cost de 118,5 milioane.

● *Etapa finală*

Dispunem de trei puncte ale curbei C_D : S_1 ($d = 26$; $c = 138$); S_3 ($d = 21$; $c \pm 127{,}5$); S_5 ($d = 23$; $c = 118{,}5$), deci o putem trasa aproximativ.

Presupunînd cunoscute curbele C_I și C_{JF}, vom determina curba costului total C_T, de unde va rezulta durata optimă de execuție a acțiunii complexe T_{opt} (fig. 4.16).

4.6. Analiza resurselor

4.6.1. Graficul Gantt asociat graficului ADC

În paragrafele precedente am arătat avantajele și limitele metodei Gantt în planificarea și conducerea acțiunilor complexe precum și avantajele metodelor ADC.

Există posibilitatea de a combina cele două metode — utilizînd de la fiecare ce are mai bun : de la metoda ADC rigurozitatea şi exactitatea, de la metoda Gantt forţa de sugestie a reprezentării. Această combinaţie ne va duce la construirea unui grafic Gantt, după ce s-a construit şi calculat graful ADC.

În figura 4.22 s-a întocmit graful Gantt asociat grafului reţea din figura 4.2. Se observă că pentru fiecare activitate în parte, intervalul în care se poate desfăşura activitatea, este reprezentat printr-o linie, adică durata între termenul minim al evenimentului de început al lucrării şi termenul maxim al evenimentului de sfîrşit.

Cu excepţia activităţilor de pe drumul critic la care durata coincide cu intervalul, pentru toate activităţile durata este mai mică decît intervalul, apărînd rezervele despre care am vorbit. Datorită rezervelor, orice activitate necritică dispune de o libertate de planificare în intervalul propriu, corelat evident cu activităţile precedente şi succedente.

4.6.2. Problema generală a resurselor în eşalonarea activităţilor unei acţiuni complexe

Pentru realizarea unei acţiuni complexe sînt necesare materii prime, materiale, utilaje, muncitori de diverse calificări, ingineri şi tehnicieni, fonduri băneşti etc. Utilizarea în timp a acestora, pe care le vom numi resurse, constituie obiectul unui capitol de maximă importanţă în metodele ADC.

Şi în metoda Gantt avem posibilitatea să examinăm necesarul de resurse şi să construim diagrame de necesar.

Astfel, să presupunem că dorim să cunoaştem necesarul pentru resursa „total muncitori" şi că din calcule de normare am determinat necesarul zilnic pe activităţi după cum urmează :

Pentru activitatea „organizare şantier" : *100* muncitori;

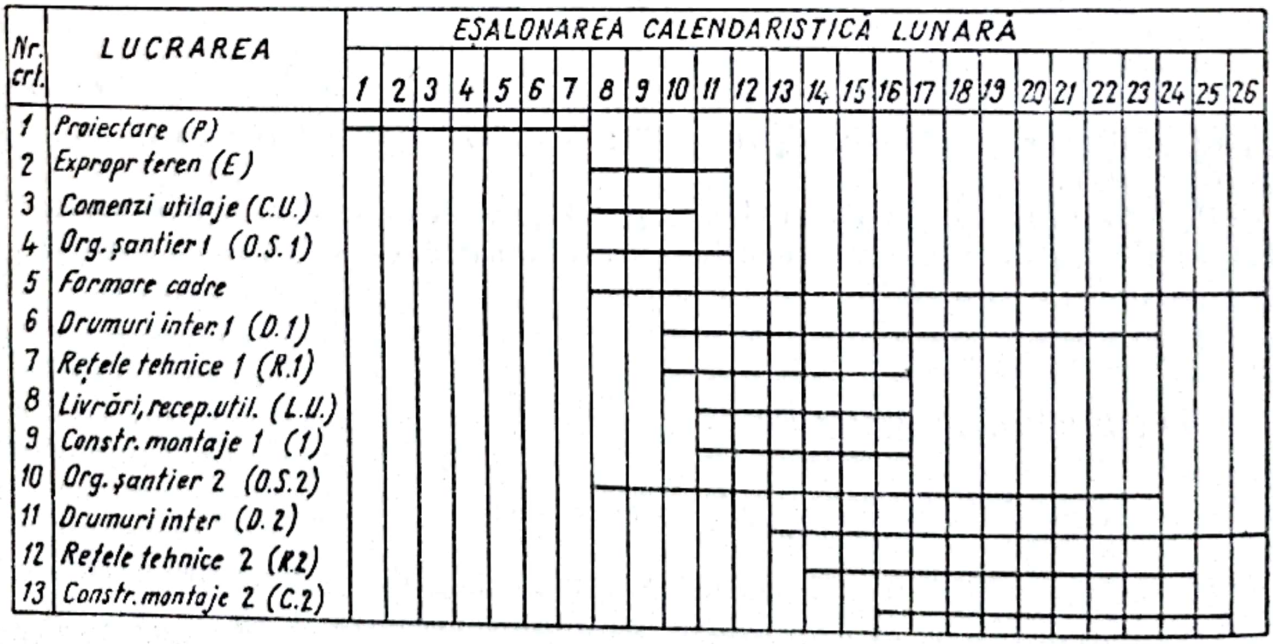


Fig. 4. 22

Scanned with OKEN Scanner

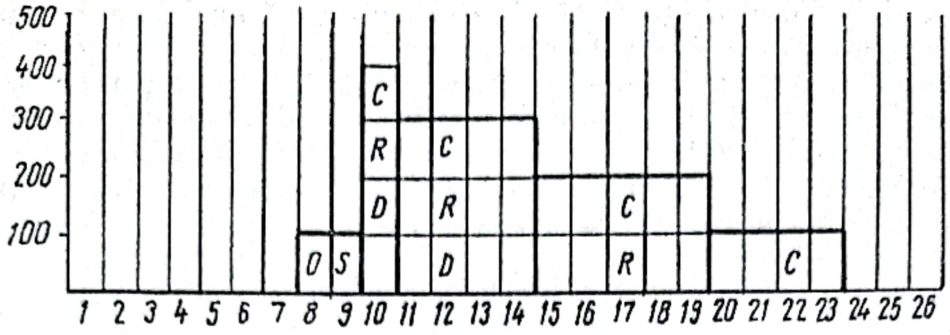


Fig. 4.23

Pentru activitatea „drumuri" : *100* muncitori;
Pentru activitatea „reţele tehnice" : *100* muncitori :
Pentru activitatea „construcţii montaj" : *200* muncitori;
pe baza graficului Gantt din figura 4.1. vom construi diagrama (fig. 4.23) necesarului total de muncitori.

Din examinarea diagramei remarcăm neuniformitatea eşalonării în timp a numărului total de muncitori, care variază în trepte de la 100 (lunile *8*, *9*, *20* — *23*) pînă la 400 (luna *10*). Eşalonarea este însă rigidă şi obligatorie, metoda Gantt nepermiţînd îmbunătăţirea ei.

Spre deosebire de această situaţie, existenţa rezervelor în graful Gantt asociat graficului ADC, permite „glisarea" unora dintre activităţi în interiorul intervalului determinat prin calcul şi înscris în grafic (fig. 4.22). Respectînd cu stricteţe intervalele şi condiţionările se pot obţine diagrame considerabil aplatisate, după cum se observă în figura 4.24.

Avînd la bază posibilitatea de a îmbunătăţi eşalonarea resurselor prin glisări în interiorul intervalelor aferente, în cadrul metodelor ADC se formulează două probleme privind resursele :

a) *Problema alocării :* considerînd pentru fiecare resursă în parte un disponibil stabilit în timp, să se determine durata minimă în care se poate executa acţiunea complexă.

d) *Problema nivelării :* considerînd termenul final obligatoriu şi egal cu durata drumului critic, să se planifice activităţile (în limita intervalelor aferente), astfel încît diagrama fiecărui necesar de resurse să fie cît mai „uniformă" (aplatisată).

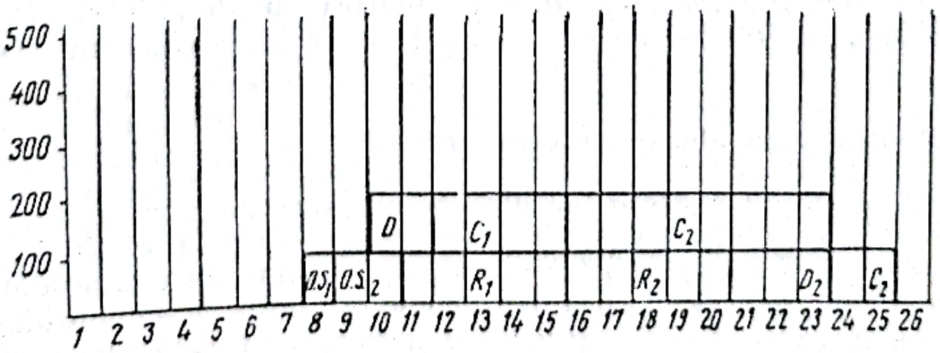


Fig. 4.24

Scanned with OKEN Scanner

Dată fiind importanța fiecăreia din cele două probleme enunțate mai sus, în cele ce urmează le vom examina mai în detaliu pe fiecare în parte. Va fi necesar, pentru început să introducem cîteva noțiuni preliminare.

Intensitatea unei resurse este cantitatea din resursa considerată, necesară sau disponibilă la un moment dat. Întrucît în metodele ADC se face ipoteza că intensitatea unei resurse este constantă pe parcursul desfășurării unei activități se poate scrie

$$K_{ij}^{n} = \frac{C_{ij}^{n}}{d_{ij}} = \frac{Q_{ij} \cdot c_{ij}^{n}}{d_{ij}}$$

în care:

K_{ij}^{n} = intensitatea resursei n pentru activitatea $i-j$;
d_{ij} = durata activității $i-j$;
Q_{ij} = cantitatea din activitatea $i-j$ (exprimată în unități naturale);
c_{ij}^{n} = consum specific (unitar) din resurse n pentru activitatea $i-j$.

Exemplu: Activitatea D_1 (execuția drumuri interioare-etapa 1) din graful-rețea construit în exemplul nostru de lucru are următoarele elemente caracteristice:

$d_{47} = 2$ luni
Q_{47} = lungimea drumului (etapa 1) = 2 km;
Să presupunem că luăm în considerare două resurse;
r_1 = fonduri bănești;
r_2 = forța de muncă totală.

Consumurile specifice pentru aceste două resurse sînt conform estimărilor tehnico-economice;

$c_{47}^{1} = 1000$ mii lei/km drum;
$c_{47}^{2} = 20000$ ore muncitor/km drum.

Cu aceste elemente vom putea calcula intensitatea celor două resurse:

$$K_{47}^{1} = \frac{2 \cdot 1000}{2} = 1000 \text{ mii lei}$$

$$K_{47}^{2} = \frac{2 \cdot 20000}{2} = 20000 \text{ ore.}$$

Profilul disponibilului unei resurse este o diagramă în care se figurează variația intensității disponibile a unei resurse în timp.

Profilul necesarului are aceeași semnificație, referindu-se însă la intensitatea necesarului unei resurse. Figurile 4.23 și 4.24 reprezintă, ambele, profilul necesarului resursei „total muncitori" în două variante de programare a activităților acțiunii complexe ce face obiectul exemplului nostru.

4.6.3. Problema alocării resurselor

Ca un exemplu sugestiv de alocare a resurselor poate fi considerată chiar problema prezentată la începutul punctului 4.6.2 în formularea căruia trebuie să adăugăm că profilul disponibilului este cunoscut. Astfel, dacă vom considera că intensitatea disponibilului resursei considerate este constantă în timp și egală cu 200 muncitori, vom avea următorul profil al disponibilului (fig. 4.25.).

Scanned with OKEN Scanner

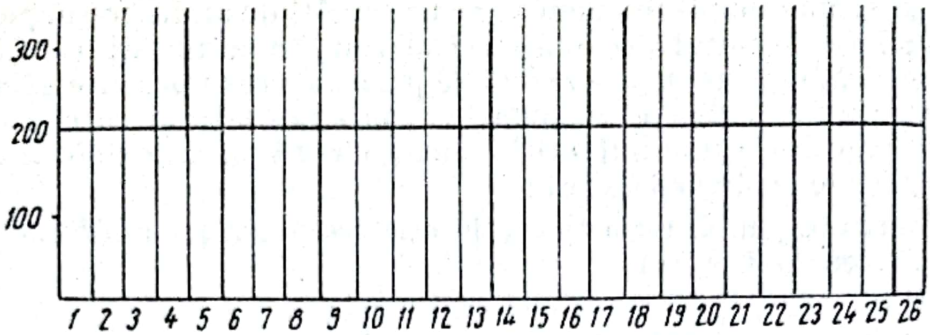


Fig. 4.25

Este evident că profilul necesarului elaborat după metoda Gantt (fig. 4.23) nu se încadrează în profilul disponibilului, prezentînd depășiri pe perioada de la 10 la 14 luni.

În schimb profilul necesarului obținut prin glisarea unor activități în limita rezervelor de care dispun (fig. 4.24) se încadrează în limita disponibilului pe tot intervalul lucrării.

Soluția satisfăcătoare pe care am obținut-o atît de ușor pentru problema noastră de alocare, prin simple încercări, se explică prin numărul redus de activități și o singură resursă luată în considerare. În practică, problemele au de cele mai multe ori sute sau chiar mii de activități și este necesară luarea în considerare a zeci de resurse importante, ceea ce face imposibil rezolvarea problemei prin mijloace empirice, de tipul încercărilor de a glisa activitățile „după ochi", ci obligă la căutarea unor metode riguroase, programabile pe calculator.

Formularea riguros-matematică a problemei alocării resurselor conduce la modele de dimensiuni și complexități foarte mari, imposibil de rezolvat chiar cu calculatoarele cele mai puternice.

Din aceste motive se utilizează procedee heuristice, care fără a da întotdeauna soluția optimă, oferă soluții cel puțin satisfăcătoare.

În cele ce urmează vom examina unele aspecte ale folosirii procedeelor heuristice de rezolvare a problemelor de alocare a resurselor.

Mersul operațiilor este în general următorul * :

a) se rezolvă problema ADC-timp, construindu-se rețeaua corespunzătoare acțiunii complexe considerate și calculînd termenele activităților, rezervele și drumul critic;

b) se trasează profilul disponibilului resurselor considerate;

c) începînd cu activitățile care încep la termenul minim de începere zero și apoi în ordinea crescătoare a termenelor minime de începere, se examinează posibilitatea de a programa aceste activități, astfel ca să nu apară depășiri ale necesarului față de disponibil, pentru nici o resursă;

d) cînd se ajunge în situația că la un anumit moment apare o astfel de depășire se încearcă rezolvarea ei prin operații de „amînare" a unora din activități (evident, nu totdeauna acest procedeu dă rezultate : este posibil ca necesarul dintr-o resursă pentru o anumită activitate să depășească el singur disponibilul; în acest caz problema alocării nu are soluții);

* În prezentarea noastră ne referim la rezolvarea mintală a problemelor, evident însă că toate operațiile se transferă corespunzător sistemelor de calcul (dealtfel mintal nu se pot rezolva decît probleme foarte mici, cu caracter de exemplu).

Scanned with OKEN Scanner

e) cînd la un anumit moment apar necesități de amînare și operația de amînare poate fi aplicată la două sau mai multe activități care încep la același termen, se introduce o *regulă de prioritate* care permite să se stabilească care anume dintre activități se programează și care se amînă; teoria și practica drumului critic menționează următoarele *criterii de amînare* folosite de diverși autori:

— prioritatea după rezerva totală cea mai mică (se amînă activitatea cu rezerva cea mai mare);

— prioritatea după durata cea mai mică;

— prioritatea după termenul minim de începere (se preferă activitatea cu t^{mi} cel mai mic);

— prioritatea după termenul maxim de începere (se preferă activitatea cu t^{Mi} cel mai mic);

— prioritatea după termenul maxim de terminare (se preferă activitatea cu t^{Mi} cel mai mic);

— prioritatea după intervalul corespunzător activității (se preferă activitatea i cu $t^{Mi} - t^{mi}$ minim).

Nu se poate vorbi despre o concluzie generală privind valabilitatea unora sau altora dintre criteriile de amînare deoarece, în anumite cazuri, apare eficientă folosirea unuia dintre ele, în alte cazuri a altuia etc.

f) se continuă procedeul pînă cînd toate activitățile sînt programate și toate resursele alocate nu depășesc disponibilul; în multe cazuri sînt necesare amînări care conduc chiar la depășirea termenului final al proiectului calculat în faza ADC-timp; ca urmare, în aceste cazuri *noțiunea de drum critic suferă o modificare devenind echivalentul duratei minime în care poate fi executată o acțiune complexă în limita resurselor disponibile.*

În general, procedeele heuristice de rezolvare a problemei alocării resurselor iau în considerare durate fixe pentru activități și nu admit întreruperea acestora. Există însă și procedee care recomandă scurtarea (sau lungirea) duratelor, după nevoie, prin alocare suplimentară, sau retragere de resurse, precum și posibilitatea de a întrerupe unele activități.

În cele ce urmează vom prezenta un algoritm de alocare a resurselor utilizînd ca regulă de prioritate rezerva totală cea mai mică *.

Pentru o bună înțelegere a algoritmului paralel cu prezentarea etapelor sale, vom urmări aplicarea lor pe un exemplu de lucru redus.

Fie acțiunea complexă definită de activitățile din tabelul 4.7.

Disponibilul pentru cele două resurse este $D_{R1} = 3$ și $D_{R2} = 4$.

Etapa preliminară

Se construiește graficul-rețea corespunzător datelor din tabel, determinîndu-se termenele activităților, rezervele totale și drumul critic (fig. 4.26).

---

* Algoritmul, în această formă, a fost elaborat de Gh. Boldur pentru cursul de cercetare operațională predat la Facultatea de Cibernetică Economică în anul universitar 1970—1971.

Scanned with OKEN Scanner

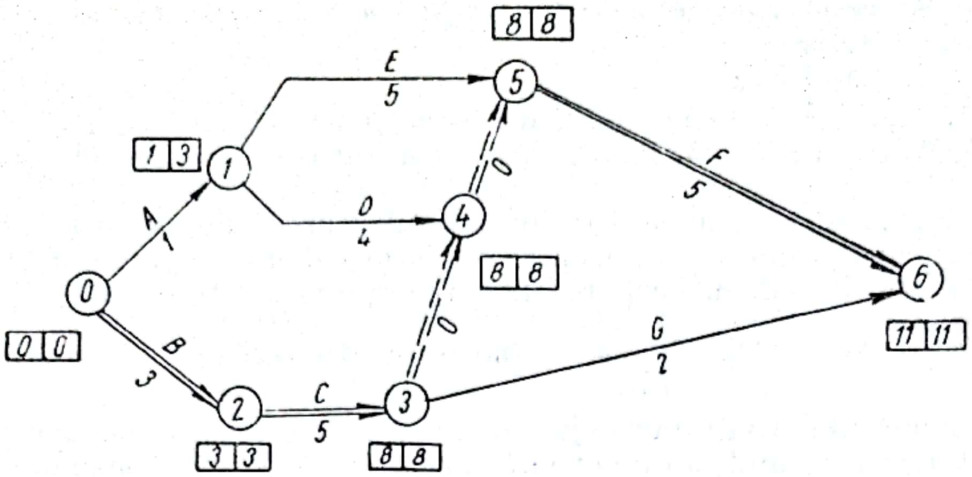


Fig. 4.26

Tabelul 4.7.

Drum	Durata	Condiționări	Necesar resurse	
			R_1	R_2
A	1	—	2	2
B	3	—	2	3
C	5	B	1	3
D	4	A	1	2
E	5	A	1	1
F	3	C,D,E	3	3
G	1	C	1	2

Termenele minime de începere ale activităților care rezultă din graficul-rețea întocmit se înscriu într-un tabel (tabelul 4.8).

Tabelul 4.8.

Nr. crt.	Denumirea activității	Termenul minim de începere (actualizare)
1	A	0 3
2	B	0
3	C	3
4	D	~~1~~ ~~4~~
5	E	~~1~~ ~~4~~
6	F	~~8~~ ~~9~~ 12
7	G	~~8~~ 9

Scanned with OKEN Scanner

Etapa de programare a activităților ținînd seama de disponibilul și necesarul de resurse.

Se introduc notațiile:

$\mathbf{D}_h (h = 1, 2, \ldots, r)$ = disponibilul din resursa h;

$r_{hi} (h = 1, 2, \ldots, r;\ i = 1, 2, \ldots, m)$ = necesarul din resursa h pentru activitatea i;

$t_j (j = 1, 2, \ldots, n)$ = termenele de începere distincte, ale activităților (se includ termenele minime de începere conform calculului graficului rețea precum și termenele de începere recalculate în urma unor amînări);

$$\Delta_{jhk} = \mathbf{D}_h - \sum_{i \in A_j} r_{hi} = \text{indicator de amînare};$$

(k = numărul activităților deja amînate dintre cele care pot începe la termenul t_j; A_j = mulțimea activităților care fie că au început la un termen anterior lui t_j dar nu s-au terminat încă, fie sînt candidate să înceapă la t_j).

● *Operația 1*

Se alege termenul minim de începere cel mai mic (din tabelul 8), fie el t_1 (în problema noastră $t_1 = 0$).

Se aleg din tabelul 4.8, activitățile care corespund lui $t_1 = 0$ în cazul nostru, activitățile **A** și **B** ($\mathbf{A}_1 = \{\mathbf{A}, \mathbf{B}\}$).

● *Operația 2*

Se scrie expresia indicatorului Δ_{jhk} pentru activitățile care pot începe la termenul t_1;

$\Delta_{110} = 3 - (2 + 2) = -1 < 0$

$\Delta_{120} = 4 - (1 + 3) = 0.$

Dacă valorile indicatorului Δ_{jhk}, scrise pentru termenul t_1 și toate resursele sînt nenegative, putem programa să înceapă la t_1 activitățile aparținînd mulțimii $\mathbf{A}_1$.

Cînd cel puțin unul dintre indicatorii Δ_{jhk} este negativ, este necesară amînarea uneia sau mai multor activități dintre cele care ar fi putut începe la termenul t_1 (în cazul exemplului $\Delta_{110} < 0$, deci avem *amînare*).

● *Operația 3*

Introducem drept criteriu de amînare „*Rezerva totală convențională** $\mathbf{R}_i = t_i^{Mt} - t_i^{mi} - d_i$, pentru fiecare activitate. (Am denumit-o rezerva totală convențională, întrucît la calculul ei se ține seama de eventualele amînări suferite de activitatea i în prealabil, adică t_i^{mi} este actualizat; din această cauză valoarea $\mathbf{R}_i$ poate fi uneori negativă.)

Din activitățile candidate să înceapă la termenul t_1, dar care nu pot fi programate toate la acest termen întrucît unul din indicatorii Δ_{jhk} este negativ, vom amîna activitatea căreia îi corespunde un $\mathbf{R}_i$ maxim (în valoare relativă).

* Acest criteriu fiind heuristic conduce în general la soluții bune, dar nu neapărat optime. În problemele mici, la care algoritmul se aplică mintal, unele observații directe pot conduce la îmbunătățirea soluției.

Scanned with OKEN Scanner

Pentru exemplul nostru avem :

$R_1 = 3 - 0 - 1 = 2$

$R_2 = 3 - 0 - 3 = 0$

deci, *se amînă activitatea* **A**.

● *Operaţia 4*

Identic cu operaţia 2, scrierea indicatorilor Δ_{ghk} făcîndu-se însă numai pentru mulţimea activităţilor rămase candidate a începe la termenul t_1, după ce a fost amînată activitatea cu $\mathbf{R}_i$ maxim

$\Delta_{111} = 3 - 2 = 1 > 0$

$\Delta_{121} = 4 - 3 = 1 > 0$

(observăm că indicatorul k a devenit *1*, ceea ce arată că *una* dintre activităţile candidate iniţial să înceapă la termenul t_1 a fost amînată).

Întrucît atît Δ_{111} cît şi Δ_{121} sînt nenegativi, rezultă *că activitatea* $\mathbf{B}_1$ *poate fi programată la termenul* t_1.

Fiind programată, activitatea **B** va fi ştearsă din tabelul 2.8 cu o linie orizontală.

● *Operaţia 5*

Amînarea unei activităţi (ca urmare a aplicării criteriului de amînare) se face conform indicaţiilor care urmează :

Fie $A^* = \{a_1, a_2, \ldots, a_p\}$ mulţimea activităţilor programate deja conform regulilor expuse, respectiv la termenele $t_{a_1}, t_{a_2}, \ldots, t_{a_p}$.

Fie a_s o activitate amînată.

Termenul de începere la care se va încerca programarea activităţilor a_s va fi

$$t^*_{a_s} = \min\ [(t_{a_1} + d_{a_1}),\ (t_{a_2} + d_{a_2}), \ldots,\ (t_{a_p} + d_{a_p})]$$

în care d_{a_g} $(g = 1, a, \ldots, p)$ este durata activităţii a_g.

$t^*_{a_s}$ reprezintă termenul de începere recalculat al activităţii a_s şi el va fi înscris în tabelul 4.8, ştergînd termenul de începere care nu mai este valabil.

În cazul exemplului nostru vom avea evident :

$$t^*_A = 0 + 3 = 3$$

pe care l-am înscris în tabelul 4.8, barînd vechiul termen $t^{mi}_A = 0$ cu o linie oblică.

Pentru urmărirea intuitivă a operaţiilor de programare vom elabora diagramele disponibil — necesar pentru ambele resurse (profilul disponibilului este figurat punctat, iar ordinea de programare a activităţilor este indicată printr-un număr încercuit).

● *Operaţia 6.* Constă în recalcularea termenelor de începere a acelor activităţi care se vor amîna datorită amînării unor activităţi care le condiţionează :

De exemplu : în graful nostru, activităţile **D** şi **E** vor fi amînate datorită amînării activităţii **A** care condiţionează pe **D** şi **E**.

Scanned with OKEN Scanner

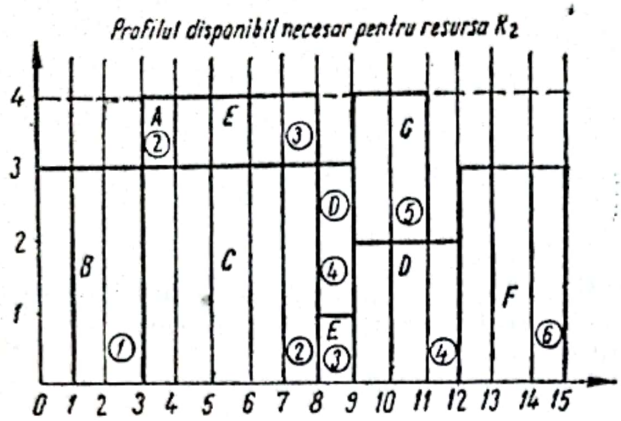


Fig. 4.27

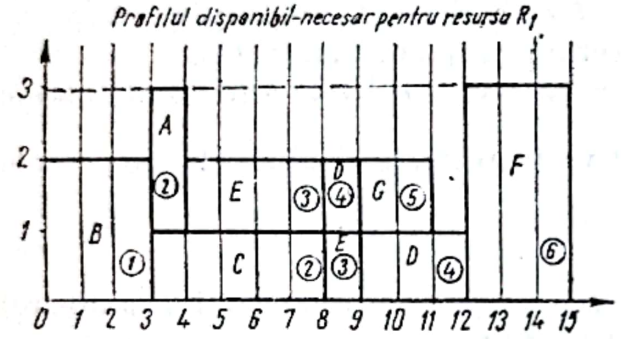


Fig. 4.28

Practic, dacă activitatea a_i este condiționată de activitățile $a_{h_1}, a_{h_2}, \ldots, a_{h_M}$, care au suferit o amînare, termenul de începere la care se va încerca programarea ei, va fi:

$$t^{**}_{a_i} = \max\,[(t^{*}_{a_{h_1}} + d_{a_{h_1}}),$$

$$(t^{*}_{a_{h_2}} + d_{a_{h_2}}), \ldots, (t^{*}_{a_{h_M}} + a_{a_{h_M}})],$$

în care $t^{*}_{a_{h_1}}, \ldots, t^{*}_{a_{h_M}}$ sînt termenele de începere recalculate ale activităților $a_{h_1}, a_{h_2}, \ldots, a_{h_M}$.

În exemplul nostru, activitățile **D** și **E** vor fi amînate după cum urmează:

$$t^{**}_{\mathbf{D}} = \max\,[(3 + 1)] = 4$$

$$t^{**}_{\mathbf{E}} = \max\,[(3 + 1)] = 4.$$

De asemenea, $t^{**}_{\mathbf{F}} = \max\,[(4 + 4), (4 + 5)] = 9$, termene recalculate pe care le înscriem în tabelul 4.8.

● *Operația 7*

Se reia algoritmul de la operația 1, continuîndu-se programarea tuturor activităților.

În cele ce urmează vom urmări aplicarea în continuare a algoritmului pentru problema care face obiectul exemplului nostru.

$.t_2 = 3,\ \mathbf{A}_2 = \{\mathbf{A}, \mathbf{C}\}$

$\Delta_{210} = 3 - (2 + 1) = 0$

$\Delta_{220} = 4 - (1 + 3) = 0.$

Deci se programează activitățile **A** și **C** la termenul $t_2 = 3$ (vezi fig. 4.27, 4.28). Aceste două activități se șterg din tabelul 4.8.

$.t_3 = 4:\ \mathbf{A}_3 = \{\mathbf{C}, \mathbf{D}, \mathbf{E}\}$

$\Delta_{310} = 3 - (1 + 1 + 1) = 0$

$\Delta_{320} = 4 - (3 + 2 + 1) = -2$ — amînare

$$\left.\begin{aligned} \mathbf{R}_4 = \mathbf{R}_{\mathbf{D}} &= \mathbf{S} - 4 - 4 = 0 \\ \mathbf{R}_5 = \mathbf{R}_{\mathbf{E}} &= \mathbf{S} - 4 - 5 = -1 \end{aligned}\right\} \text{se amînă } \mathbf{D}.$$

$\Delta_{311} = 3 - (1 + 1) = 1 > 0$

$\Delta_{321} = 4 - (3 + 1) = 0.$

Scanned with OKEN Scanner

Deci se programează activitatea **E** la termenul $t_3 = 4$.
Se amînă **D** la termenul :

$$t_D^* = \min\ [(3 + 5),\ (4 + 5)] = 8$$

(se operează modificarea în tabelul 4.8).
Se amînă **F** la termenul

$$t^{**} = \max\ [(8 + 4)] = 12$$

$$t_4 = 8;\ \mathbf{A}_4 = \{\mathbf{E}, \mathbf{D}, \mathbf{G}\}$$

$$\Delta_{410} = 3 - (1 + 1 + 1) = 0$$

$$\Delta_{420} = 4 - (1 + 2 + 2) = -1 - \text{amînare}$$

$$\left.\begin{aligned} \mathbf{R}_D &= 8 - 8 - 4 = 4 \\ \mathbf{R}_G &= 11 - 8 - 2 = 1 \end{aligned}\right\} \text{se amînă } \mathbf{G}$$

$$\Delta_{411} = 3 - (1 + 1) = 1$$

$$\Delta_{421} = 4 - (1 + 2) - 1.$$

Se programează activitatea **D** la termenul $t_4 = 8$.
Se amînă **G** la termenul

$$t_G^* = \min\ [(4 + 5),\ (8 + 4)] = 9$$

$$t_5 = 9;\ \mathbf{A}_5 = \{\mathbf{D}, \mathbf{G}\}$$

$$\Delta_{510} = 3 - (1 + 1) = 1$$

$$\Delta_{520} = 4 - (2 + 2) = 0.$$

Se programează **G** la termenul $t_5 = 9$

$$t_6 = 12;\ \mathbf{A} = \{\mathbf{F}\}$$

$$\Delta_{610} = 3 - 3 = 0$$

$$\Delta_{620} = 4 - 3 = 1.$$

Se programează **F** la termenul $t_6 = 12$.
Observăm că termenul de execuţie al lucrării a crescut de la 11 unităţi de timp (fig. 4.26), la 15 unităţi de timp (fig. 4.27 şi 4.28).

4.6.4. Problema nivelării resurselor

După cum am arătat această problemă constă în planificarea activităţilor, cu limitarea termenului final de execuţie a acţiunii complexe, astfel încît profilul necesarului resurselor să fie cît mai uniform.

Există mai multe moduri de a exprima obiectivul uniformizării profilului necesarului, putîndu-se urmări :

— minimizarea sumei variaţiilor absolute din profil ;
— minimizarea sumei creşterilor ;

Scanned with OKEN Scanner

— minimizarea deviațiilor absolute de la medie;
— minimizarea valorii maxime;
— minimizarea variației maxime;
— minimizarea sumei pătratelor diferențelor între profilul necesarului și un profil ideal (de obicei o linie paralelă cu axa timpului) etc.

Utilizînd criteriul minimizării sumei pătratelor diferențelor, Burgess și Killebrew au elaborat un algoritm de nivelare pentru o singură resursă ale cărui operații sînt următoarele:

a. Se întocmește graficul-rețea și se calculează drumul critic;

b. Se programează activitățile la termenul minim de începere, se întocmește profilul necesarului și se calculează suma pătratelor diferențelor (pe fiecare unitate de timp) între profilul necesarului și profilul disponibilului;

c. Începînd de la sfîrșitul graficului-rețea se ia prima activitate care dispune de rezervă și se încearcă care este poziția cea mai avantajoasă posibilă a ei, din punct de vedere a criteriului enunțat; cînd sînt mai multe poziții egal avantajoase se alege cea mai din dreapta.

d. Se trece la următoarea activitate, spre început, care dispune de rezervă și se procedează la fel apoi, pentru toate activitățile;

e. Se calculează valoarea criteriului corespunzător noii planificări obținute apoi se aplică din nou algoritmul de la pasul *c*, pînă ce nu mai sînt posibile îmbunătățiri.

În cazul problemelor de nivelare după mai multe resurse este posibilă adaptarea algoritmului Burgers-Killebrew după cum urmează:

— se ierarhizează importanța relativă a resurselor stabilindu-se niște coeficienți de pondere;

— se începe aplicarea algoritmului, urmărindu-se simultan pentru toate resursele efectului deplasării unei activități în limita intervalului ei aferent; se pot ivi următoarele cazuri;

1) deplasarea unei activități îmbunătățește valoarea criteriului pentru toate resursele considerate, în acest caz nu se face nici un calcul, deplasarea efectuîndu-se cît mai avantajos;

2) deplasarea unei activități îmbunătățește valoarea criteriului pentru o parte din resurse și o înrăutățește pentru altele dintre ele; în acest caz se urmărește acea poziție a activităților care face ca suma sumei pătratelor diferențelor pentru fiecare resursă, ponderată cu coeficienții de pondere stabiliți, să fie minimă.

4.7. Alte procedee de analiză a drumului critic

4.7.1. Metoda potențialelor

Spre deosebire de metoda CPM (cu reprezentare pe arce și noduri), în metoda potențialelor, se admite că o activitate poate începe înainte de terminarea activităților care o condiționează și reciproc se poate termina după începerea activităților pe care le condiționează. Această convenție

Scanned with OKEN Scanner

echivalează cu introducerea unui nou tip de condiţionare : începerea unei anumite activităţi **B** este condiţionată de executarea unei părţi a activităţii **A** care se poate considera divizată în două părţi componente $\mathbf{A}_1$ şi $\mathbf{A}_2$ (fig. 4.29).

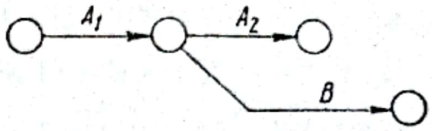


Fig. 4.29

Pentru a putea calcula termenele de începere şi terminare ale activităţii **B** va fi necesar deci să cunoaştem durata $d_{\mathbf{A}_1}$ a activităţii $\mathbf{A}_1$ (fig. 4.29). Întrucît în metoda potenţialelor noi nu divizăm activitatea **A**, va fi necesar să introducem pe altă cale durata activităţii $\mathbf{A}_1$. Acest lucru se realizează reprezentînd activităţile pe noduri şi asociind arcelor durate avînd semnificaţia unor durate parţiale de activităţi.

În figura 4.30 se arată reprezentarea activităţilor în metoda potenţialelor. Se observă că s-au asociat arcelor mărimile t_{ij} care au semnificaţia intervalul de timp de la începerea activităţii i, pînă cînd poate începe activitatea j, adică exact durata de tipul $d_{\mathbf{A}_j}$, din exemplul de mai sus. Calculul termenelor minime şi a rezervelor diferă faţă de metoda CPM (reprezentarea pe noduri) şi se face cu ajutorul formulelor :

$$t^i_{mi} = \max\ [(t^{h_1}_{mi} + t_{h_1 i}),\ (t^{h_2}_{mi} + t_{h_2 i}), \ldots, (t^{h_m}_{mi} + t_{h_m i})]$$

$$t^i_{Mi} = \min\ [(t^{j_1}_{Mi} - t_{ij_1}),\ (t^{j_2}_{Mi} - t_{ij_2}), \ldots, (t^{j_n}_{Mi} - t_{ij_n})]$$

$$t^i_{mt} = t^i_{mi} + d_i$$

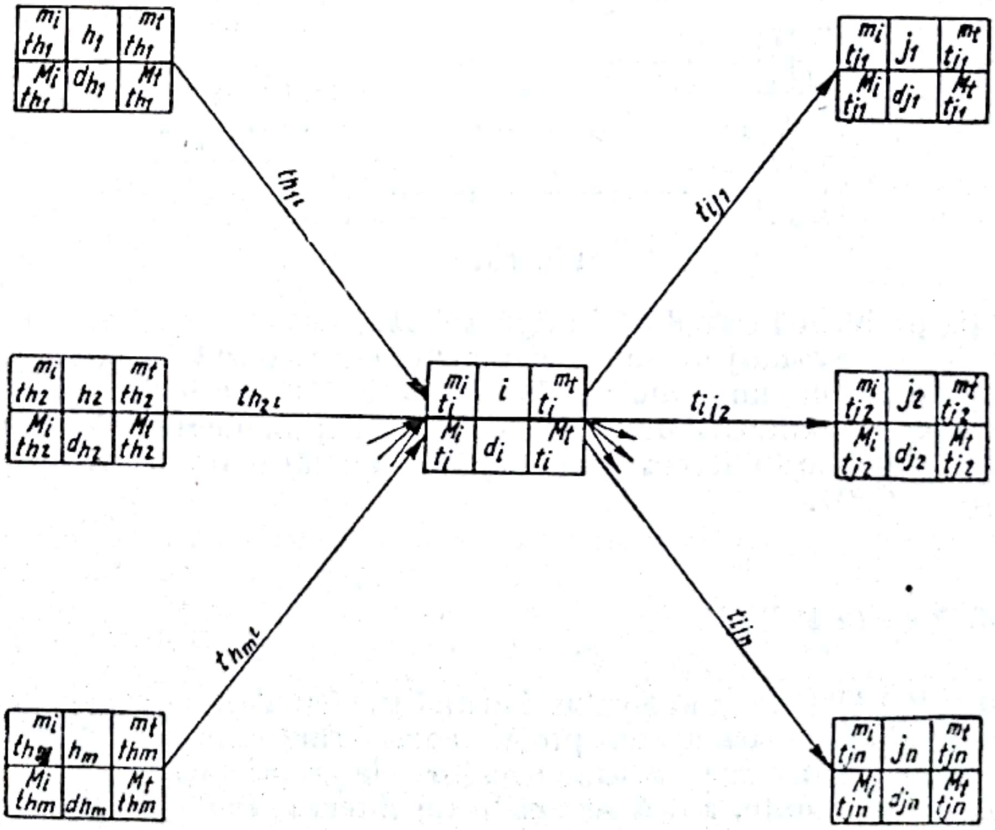


Fig. 4.30

$$t^i_{Mi} = t^i_{Mi} + d_i \neq \min\ [t^{h_1}_{Mi}\ t^{h_2}_{Mi}, \ldots,\ t^{h_m}_{Mi}]$$

$$R_{ii} = t^i_{Mi} - t^i_{mi} - d_i = t^i_{Mi} + d_i - t^i_{mi} - d_i = t^i_{Mi} - t^i_{mi}.$$

Exemplu: Să se construiască graficul-reţea definit de elementele din tabelul 4.9 şi să se calculeze termenele şi drumul critic în metoda potenţialelor.

Tabelul 4.9

Nr. crt.	Denumirea activităţii	Durata	Condiţionări	Intervale t_{ij}
0	1	2	3	4
1	A	3	—	—
2	B	4	A	$t_{AB} = 2$
3	C	3	A	$t_{AC} = 2$
4	D	3	A	$t_{AD} = 3$
5	E	5	B, C	$t_{BE} = 3, t_{CE} = 3$
6	F	6	C, D	$t_{CF} = 2, t_{DF} = 2$
7	G	4	D, D	$t_{BG} = 2, t_{DG} = 2$
8	H	6	D, E	$t_{DH} = 2, t_{EH} = 4$
9	I	3	G	$t_{IG} = 3$
10	J	2	H, J	$t_{HJ} = 4, t_{IJ} = 2$

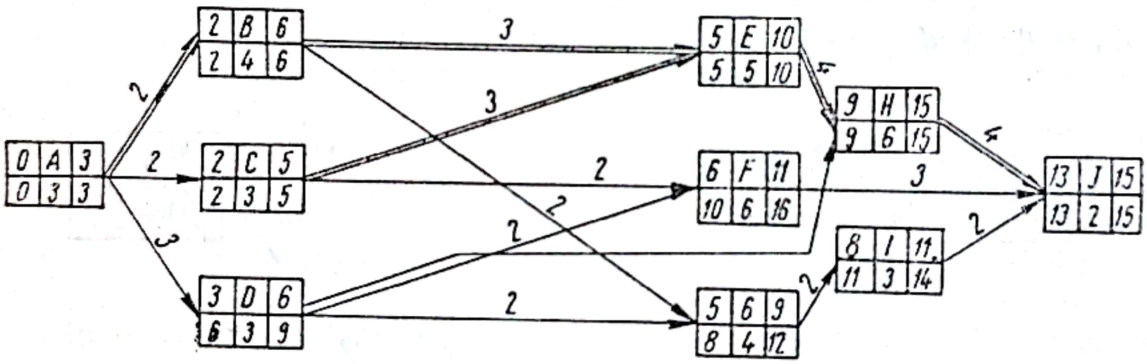


Fig. 4.31

Soluţia problemei este dată în figura 4.31.

Principalul avantaj al metodei potenţialelor constă în posibilitatea de a reprezenta cu un număr redus de activităţi, acţiunile complexe. Într-adevăr, posibilitatea de a reprezenta „suprapunerea" în timp a activităţilor, dă posibilitatea de a diminua numărul de activităţi, faţă de metoda CPM.

4.7.2. Metoda PERT

Metoda PERT (Program Evaluation and Review Technique) analizează statistic durata estimată a unui proiect considerînd duratele activităţilor ca variabile aleatoare care satisfac următoarele proprietăţi :

1. Să fie mărginite, adică să existe un interval finit [A, B] în care se ia valori.

Scanned with OKEN Scanner

2. Să existe o valoare **M** cea mai probabilă, $\mathbf{M} = \in [\mathbf{A}, \mathbf{B}]$

3. $\mathbf{B} - \mathbf{A} = 6\ \sigma$.

De obicei, se consideră că duratele activităţilor sînt variabile beta. O variabilă aleatoare **V** are repartiţia beta dacă densitatea sa de repartiţie este :

$$g(x) = \begin{cases} \dfrac{1}{\mathbf{B}(p-q)} x^{p-1}(1-x)^{q-1} & x \in [0,1] \\ 0 \text{ în rest} \end{cases}$$

unde

$$\mathbf{B}(p, q) = \int_0^1 x^{p-1}(1-x)^{q-1}\, dx.$$

Variabila **V** nu satisface condiţiile *1* şi *2* de mai sus ; de aceea trebuie să facem transformarea :

$$\mathbf{W} = (\mathbf{B} - \mathbf{A})\mathbf{V} + \mathbf{A}.$$

Densitatea de repartiţie a variabilei aleatoare **W** este

$$f(x) = \begin{cases} \dfrac{1}{(\mathbf{B}-\mathbf{A})^{p+q-1}\,\mathbf{B}(p, q)} (x - \mathbf{A})^{p-1}(\mathbf{B} - x)^{q-1}, & x \in [\mathbf{A}, \mathbf{B}] \\ 0 \text{ în rest} \end{cases}$$

Variabila **W** se mai numeşte şi variabilă beta PERT. Se arată că

$$\mathbf{M} = \frac{\mathbf{A}(q-1) + \mathbf{B}(p-1)}{p+q-2}, \quad \mu = \frac{\mathbf{A} + (p+q-2)\mathbf{M} + \mathbf{B}}{p+q},$$

$$\sigma^2 = \frac{pq(\mathbf{B}-\mathbf{A})^2}{(p+q+1)(p+q)^2}.$$

Dacă se impune condiţia $\mathbf{B} - \mathbf{A} = 6\ \sigma$ şi presupunem că $p + a = 6$ avem

$$\mu = \frac{\mathbf{A} + 4\mathbf{M} + b}{6}, \quad \sigma = \frac{\mathbf{B} - \mathbf{A}}{6}$$

care se ia ca formulă pentru determinarea duratei medii μ_a şi a dispersiei σ_a^2 pentru toate activităţile proiectului.

Cu aceste durate medii se calculează drumul critic, folosind metodele anterioare (CPM, de exemplu) şi se determină un drum crtic, notat $\mathcal{C}$. Durata totală **T** şi dispersia ei pentru proiectul respectiv se calculează cu formulele :

$$\mathbf{T} = \sum_{a \in \mathcal{C}} \mu_a, \quad \sigma_{\mathbf{T}}^2 = \sum_{a \in \mathcal{C}} \sigma_a^2$$

Scanned with OKEN Scanner

În ceea ce privește interpretarea rezultatelor problemelor PERT considerate probabilistic, admitem că se verifică teorema limită centrală, deci că durata totală **T** are o repartiție normală. Pe baza acestei ipoteze se compară durata totală **T** cu termenul planificat $\mathbf{T}_p$ calculînd abaterea normată.

$$x = \frac{\mathbf{T}_p - \mathbf{T}}{\sigma_{\mathbf{T}}}$$

și utilizînd funcția lui Laplace (repartiția normală $(0,1)$)

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-\frac{x^2}{2}} dx$$

se poate stabili probabilitatea de a respecta termenul planificat $\mathbf{T}_p$.

Într-o planificare rațională a activităților se recomandă ca probabilitatea de încadrare în termen a proiectului să oscileze în intervalul $[0,5; 0,6]$. Pentru o probabilitate mai mare decît 0,6 siguranța de realizare a termenului planificat $\mathbf{T}_p$ crește, dar este necesară o suplimentare de resurse.

Pentru aplicarea metodei PERT este necesar să fie îndeplinite și următoarele condiții esențiale:

i) lungimea drumului critic să difere semnificativ (mai mult de 15%) de lungimea oricărui alt drum complet astfel ca să se asigure un singur drum critic probabil;

ii) drumul critic trebuie să conțină un număr mare de activități fără ca nici una să fie preponderentă (durata unei activități să fie cel mult 10% din durata totală) pentru a se putea aplica teorema limită centrală.

4.8. Procedee complexe de alocare-nivelare a resurselor; sisteme de programe pentru calculator

În problemele practice, atît alocarea cît și nivelarea resurselor reprezintă obiective importante, cu mare pondere economică. Din această cauză s-au elaborat metode prin care se încearcă combinarea acelor două obiective, urmărindu-se în primul rînd încadrarea în disponibilul resurselor, iar apoi, odată realizată această încadrare, nivelarea profilului disponibilului.

Pe de altă parte, complexitatea extraordinară a calculelor, face imposibilă rezolvarea manuală a problemelor de alocare și nivelare, chiar pentru grafice-rețea cu activități puține și luînd în considerare numai 2—3 resurse.

Scanned with OKEN Scanner

Au apărut astfel sistemele de programe pe calculator care încearcă rezolvarea problemelor de alocare și nivelare pentru grafice-rețea cu multe activități și luînd în considerare zeci de resurse.

Sisteme de programe pentru analiza drumului critic. Sistemul de programe PERT—RES, elaborat în cadrul laboratoarelor Catedrei de Cibernetică Economică din A.S.E. este utilizat pentru eșalonarea lucrărilor reprezentate prin grafice rețea cu activitățile pe arce. Se calculează lungimea drumului critic și se optimizează rezervele necesare executării activităților, în sensul nivelării și alocării resurselor. Un grafic rețea poate avea maximum 3 000 activități și 50 resurse diferite, pe fiecare activitate pot fi definite maximum 8 resurse. Formularele de ieșire sînt diverse: tabele de activități, grafice Gantt, tabele de resurse, histograme de resurse. Acestea sînt sortate și paginate la cererea utilizatorului. Sistemul necesită o memorie internă de 64 K.

Pachetul de programe PERT—CORAIL — 2 utilizează grafice rețea cu activitățile pe noduri. Calculează datele de începere și terminare, minime și maxime pentru activități și nivelează încărcarea resurselor. Sînt listate toate formularele standard. Sistemul necesită o memorie internă de 128 K.

Aceste sisteme se aplică în eșalonarea activităților în revizii și reparații, în conducerea operativă a investițiilor etc.

Scanned with OKEN Scanner

Capitolul V

GESTIUNEA STOCURILOR

5.1. Introducere

Desfăşurarea în bune condiţii a activităţilor în cadrul unităţilor economice impune asigurarea unor cantităţi corespunzătoare de resurse materiale : materii prime şi materiale, combustibil, piese de schimb, semifabricate de diverse tipuri etc.

Întrucît consumul resurselor are un caracter continuu iar aprovizionarea se face de regulă la anumite momente, se pune problema determinării cantităţilor de resurse ce trebuie stocate între momentele de reaprovizionare pentru asigurarea continuităţii procesului economic.

Prin gestiunea ştiinţifică a stocurilor se urmăreşte fundamentarea deciziilor privind conducerea procesului de consum-aprovizionare. În cadrul gestiunii ştiinţifice a stocurilor un rol important îl are stabilirea pe baza unor modele economico-matematice a nivelului stocurilor din diferitele resurse şi a momentelor la care se face reaprovizionarea.

Modelele prezentate în acest capitol sînt destinate iniţierii cititorului în problemele ce fac obiectul optimizării deciziilor privind procesul de stocare. Caracteristic pentru aceste modele de stoc este faptul că fenomenul economic analizat se descrie printr-o funcţie a costului total de stocare $F(s)$ în care s reprezintă volumul de materii prime, materiale, produse finite etc. care trebuie să existe în unitate la diferite momente. Prin determinarea valorii extreme a funcţiei $F(s)$ rezultă stocul s_0 care minimizează această funcţie. Modelele din această clasă au aplicaţii în diverse domenii de activitate : producţia industrială, circulaţia mărfurilor, operaţii financiare etc.

Se poate spune că un model de stoc reprezintă un instrument cu ajutorul căruia se minimizează volumul de fonduri ce se imobilizează în procesul de stocare şi în acelaşi timp se asigură continuitatea procesului de producţie sau desfacere. Trebuie precizat că în aplicaţiile ce se referă la satisfacerea cererii consumatorilor se poate pune problema maximizării venitului net ce rezultă din activitatea de desfacere a produselor. Principalele elemente care intervin într-o gestiune de stocuri sînt :

1. Cererea de articole notată **U** şi care poate fi deterministă sau aleatoare ;

Scanned with OKEN Scanner

2. Perioada de livrare notată τ, care şi ea poate fi deterministă, poate să depindă de volumul comenzii de aprovizionare sau poate fi aleatoare;

3. Nivelul stocului.

În funcţie de condiţiile concrete ce caracterizează fenomenul analizat se pot lua în considerare următoarele niveluri de stoc : nivel maximal, capacitatea limită de stocaj, nivelul minimal, nivelul de alarmă, nivel de reaprovizionare;

4. Costurile utilizate în modele de stoc :

a) costul de stocaj c_s se consideră pe unitatea de produs stocat în unitate de timp şi cuprinde cheltuielile legate de stocarea materialelor sau produselor : cheltuieli pentru depozitare, întreţinere, reparaţii, valoarea produselor perisate în timpul stocării, imobilizarea fondurilor băneşti etc.

b) costul de lansare c_1, reprezintă cheltuielile administrative legate de operaţia de reaprovizionare;

c) costul de penalizare c_p, care apare în cazul modelelor care admit ruptura de stoc se evaluează prin producţia netă nerealizată datorită lipsei de stoc (pierderi din stagnarea producţiei), amenzi, cheltuieli suplimentare pentru satisfacerea cererii etc. Se poate întîmpla ca în unele situaţii costurile de lansare şi de stocaj să fie neglijabile în care caz introducem următoarele elemente :

— o pierdere c_1 la articolele soldate atunci cînd cererea **U** este inferioară stocului S;

— un cost suplimentar c_2 pentru o aprovizionare specială atunci cînd cererea este superioară stocului S.

5. Volumul comenzii de aprovizionare n, care poate fi constant sau variabil în funcţie de o lege de gestiune pe care o adoptăm.

6. Datele la care se face aprovizionarea notate cu t, precum şi perioadele de gestiune (ciclul de reaprovizionare) pe care le notăm cu **T**.

La rîndul lor aceste elemente pot fi constante sau variabile, variaţia lor urmînd unei anumite reguli de gestiune pe care o adoptăm.

5.2. Reguli de gestiune

Evoluţia nivelului stocului în timpul unei perioade T pentru fiecare articol separat se reprezintă sub forma unei funcţii în scară, descrescătoare în timp. Din motive teoretice este comod să se înlocuiască graficul în scară printr-o dreaptă (în cazurile simple) sau printr-o curbă descrescătoare în general, care să ne dea imaginea variaţiei cererii în timp exprimată analitic, formă de exprimare mult mai comodă.

Iată mai jos cum arată un exemplu de variaţie în timp a stocului pe o perioadă T :

Există numeroase reguli de gestiune, reguli care conduc la modele matematice corespunzătoare. Vom examina pe scurt — mai bine zis schematic — cele mai importante modele şi care apar mai frecvent în aplicaţii.

Scanned with OKEN Scanner

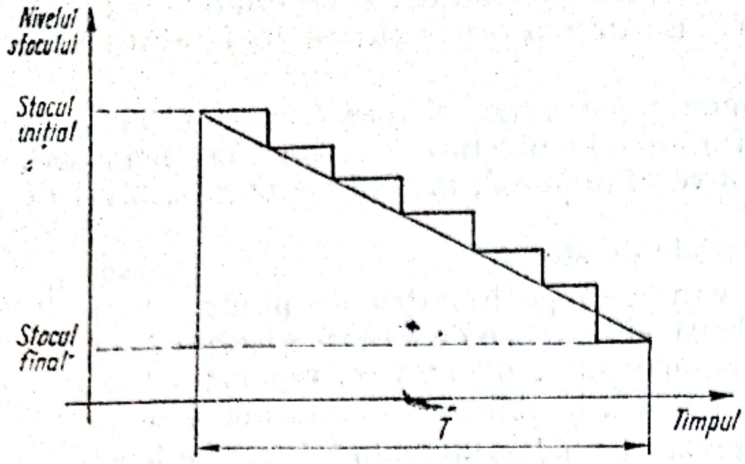


Fig.5.1

5.2.1. Gestiuni cu perioada fixă de aprovizionare

În cazul acestui tip de gestiune vom distinge cazurile :

a) Cererea este constantă, aprovizionarea are loc instantaneu și în cantități constante n ; nu există ruptura de stoc.

b) Cererea este variabilă — deterministă sau aleatoare.

Aprovizionarea se face în cantități variabile n_i, astfel încît stocul va atinge nivelul maximal la sfîrșitul fiecărei perioade, iar aprovizionarea va necesita un anumit timp constant, τ.

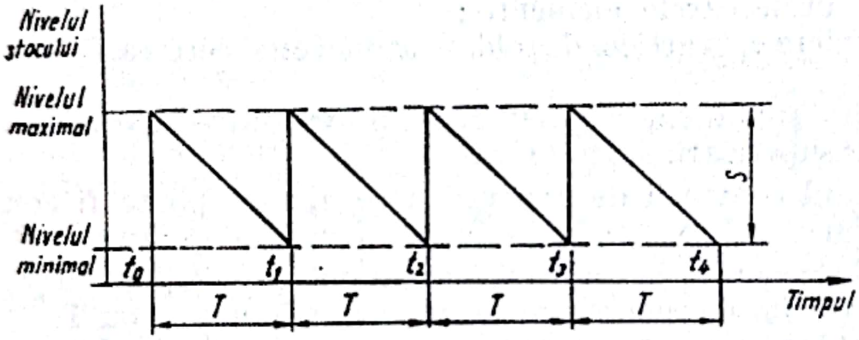


Fig. 5.2

Cantitățile n_i trebuie să fie estimate la momentele t_i prin extrapolarea dreptei sau a curbei de consum. În general în astfel de modele nivelul maximal coincide cu capacitatea limită de stocaj.

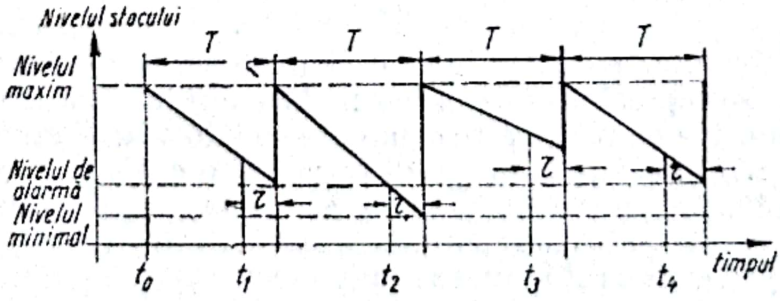


Fig. 5.3

Pe figura 5.3. se constată că avem un model cu ruperea de stoc, în care stocul devine la un moment dat nul. Nivelul de alarmă (alertă) este pragul sub care se poate produce o ruptură de stoc cu o probabilitate neglijabilă.

Scanned with OKEN Scanner

5.2.2. Gestiuni cu perioada variabilă de aprovizionare

Ca şi în situaţia precedentă vom distinge două situaţii :

a) Cererea este variabilă — deterministă sau aleatoare; aprovizionarea se face prin cantităţi fixe n, astfel că stocul va atinge nivelul maximal la

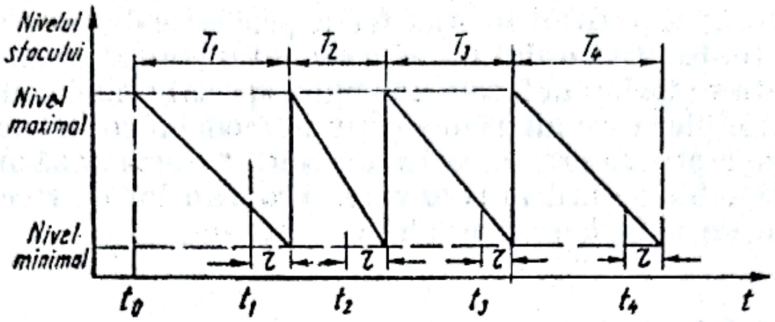


Fig. 5.4

sfîrşitul fiecărei perioade, fiind necesar (pentru aprovizionare) un anumit timp constant τ.

Datele t_i la care se lansează comenzile trebuie determinate prin extrapolarea dreptei sau curbei de consum.

Un astfel de model este reprezentat grafic în figura 5.4.

b) Cererea este variabilă — deterministă sau aleatoare —.

Aprovizionarea se face prin cantităţi fixe şi necesită un anumit timp constant de la lansarea comenzii.

Decizia de efectuare a comenzii se face în momentul în care stocul atinge un anumit nivel ales convenabil pe care-l vom numi nivel de reaprovizionare, situat deasupra nivelului de alarmă.

Grafic, acest model arată ca în figura 5.5.

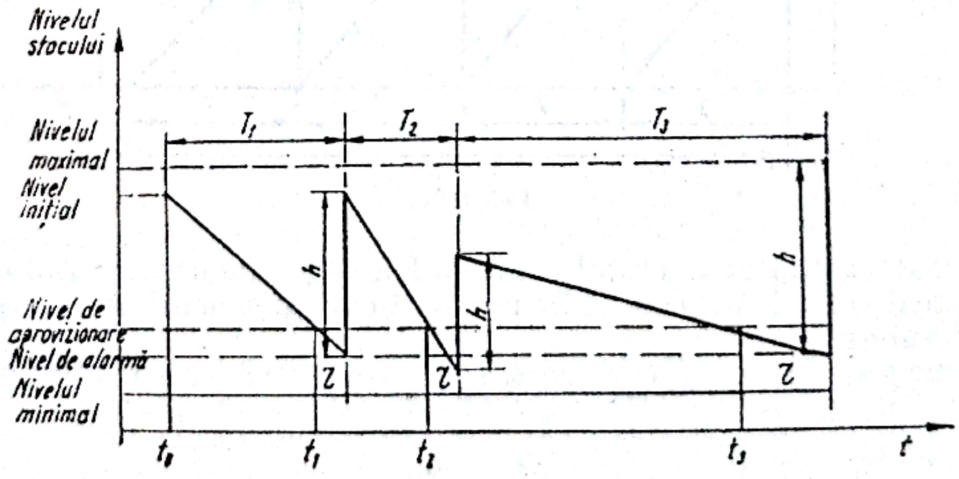


Fig. 5.5

Metoda aceasta de gestiune oferă avantajul unei gestiuni comode, dar are dezavantajul că nu garantează cu o probabilitate suficientă absenţa rupturilor de stoc.

Scanned with OKEN Scanner

5.3. Modele matematice deterministe folosite în optimizarea proceselor de stocare

În cadrul acestui paragraf se vor prezenta modelele matematice corespunzătoare diferitelor tipuri de gestiune. Subliniem faptul că în practică, așa cum se întîmplă dealtfel în orice fel de probleme de cercetare operațională, nu este totdeauna posibil să reducem cazul studiat la un model clasic și drept urmare va trebui să facem un studiu special ținînd cont de particularitățile restricțiilor care intervin efectiv în modelul considerat.

Menționăm faptul că programarea dinamică * furnizează un instrument matematic deosebit de util în rezolvarea problemelor de stocaj, care sînt esențialmente de natură secvențială.

5.3.1. Model de gestiune cu perioada fixă și cerere constantă

Într-un astfel de model, cînd nu există ruptura de stoc nu avem cheltuieli de penalizare și intervin deci numai cheltuieli legate de lansarea comenzii și cheltuieli de stocaj.

Se presupun date intervalul de timp θ pe care se face gestiunea și cercetarea totală N de produse pe acest interval de timp.

În aceste condiții se cere să se determine cantitatea n cu care se face aprovizionarea periodică după fiecare perioadă de timp T, care trebuie

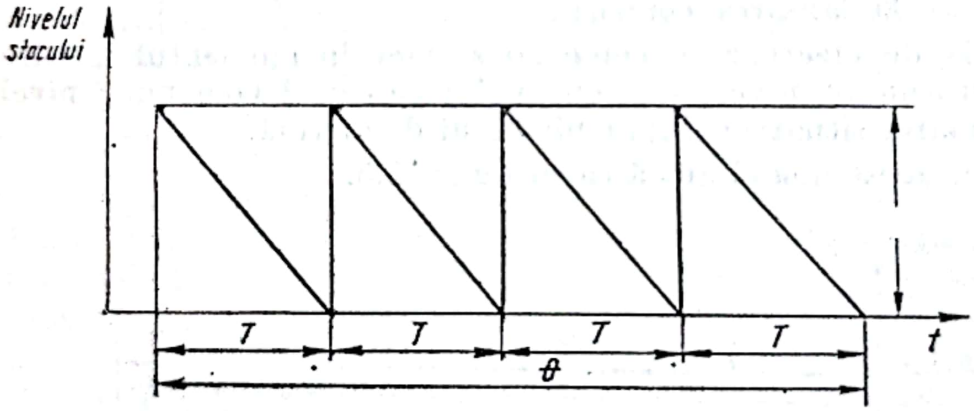


Fig. 5.6.

determinată astfel încît costul total de lansare a comenzilor și de stocaj pentru toate cele N articole să fie minim (în absența penalizărilor așa cum am presupus).

Întrucît cererea este constantă și perioada fixă putem scrie imediat egalitatea :

$$\frac{N}{n} = \frac{\theta}{T}$$

* Capitolul VI, consacrat programării dinamice, cuprinde exemple de rezolvare a problemelor de stocare.

Scanned with OKEN Scanner

de unde urmează :

$$T = \frac{n\theta}{N}.$$

Pe de altă parte costul corespunzător unei singure perioade este (cheltuielile de lansare, la care se adaugă cheltuielile de stocare) :

$$c_1 + \frac{1}{2} n T c_s$$

Cum avem $\frac{N}{n}$ perioade, rezultă că costul total pe întregul interval θ de gestiune este :

$$F(n) = \left(c_1 + \frac{1}{2} n T c_s \right) \frac{N}{n} = \frac{N c_1}{n} + \frac{\theta c_s}{2} n.$$

Dacă notăm cu $F_1(n)$ cheltuielile totale de lansare, cu $F_s(n)$ cheltuielile totale de stocare cu n_0 și F_0 mărimea optimă a unei comenzi și respectiv cheltuielile minime, atunci modelul considerat poate fi reprezentat grafic ca în figura 5.7.

Trebuie să determinăm minimul funcției $F(n)$.

Întrucît $F_s(n) \cdot F_1(n) = \text{constant}$, funcția $F(n)$ atinge minimul cînd $F_1(n) = F_s(n)$ și de aici rezulta că :

$$\frac{\theta c_s}{2} n = \frac{N c_1}{n} \text{ adică } n^2 = \frac{2 N c_1}{\theta c_s}$$

și deci :

$$n_0 = \sqrt{\frac{2 N c_1}{\theta c_s}}.$$

(Cealaltă soluție neputînd fi admisă din cauza că este negativă.)
De aici se deduce imediat :

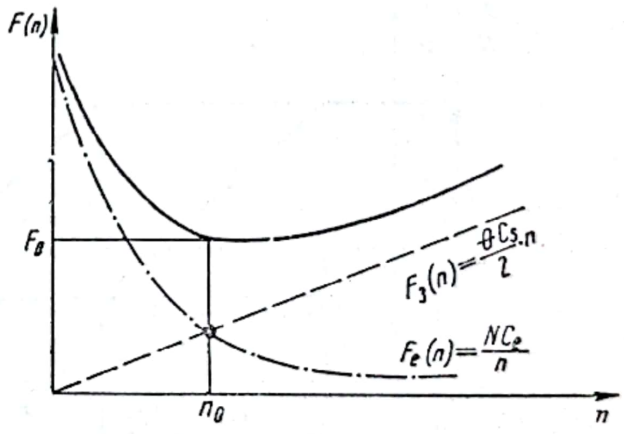


Fig. 5.7

$$T_0 = \frac{n^0 \theta}{N} = \sqrt{\frac{2\theta c_1}{N c_s}}$$

și

$$F_0 = F(n_0) = \frac{N c_1}{\sqrt{\frac{2 N c_1}{\theta c_s}}} + \frac{\theta c_s}{2} \sqrt{\frac{2 N c_1}{\theta c_s}} = \sqrt{2 N \theta c_1 c_s}$$

Scanned with OKEN Scanner

5.3.2. Model de gestiune cu perioadă fixă și cerere constantă cu posibilitățile de ruptură a stocului

Fenomenul de ruptură a stocului apare atunci cînd o cerere nu poate fi satisfăcută. Acestei lipse de stoc i se atașează costul de penalizare c_p estimat pe unitatea de produs în unitatea de timp. Vom presupune că cheltuielile de lansare c_l sînt aceleași pentru fiecare perioadă T. Schematic această situație apare ca în figura 5.8.

Pe parcursul unei durate T_1 din fiecare perioadă T nivelul stocului este suficient pentru a satisface cererea, pe parcursul duratei $T_2 = T - T_1$ avem o penurie de articole pînă la sfîrșitul perioadei T în care timp se plătește o penalizare c_p pentru fiecare articol pe unitate de timp.

În felul acesta avem următoarele cheltuieli corespunzătoare unei perioade T:

— cheltuieli de stocaj: $\frac{1}{2}\, sT_1c_s$

— cheltuieli de lansare: c_l

— cheltuieli de penalizare: $\frac{1}{2}(n-s)T_2c_p$

Costul pe intervalul de timp θ pe care se consideră gestiunea va fi:

$$F(n, s) = \left[\frac{1}{2}sT_1c_s + c_1 + \frac{1}{2}(n-s)T_2c_p\right]\frac{N}{n}$$

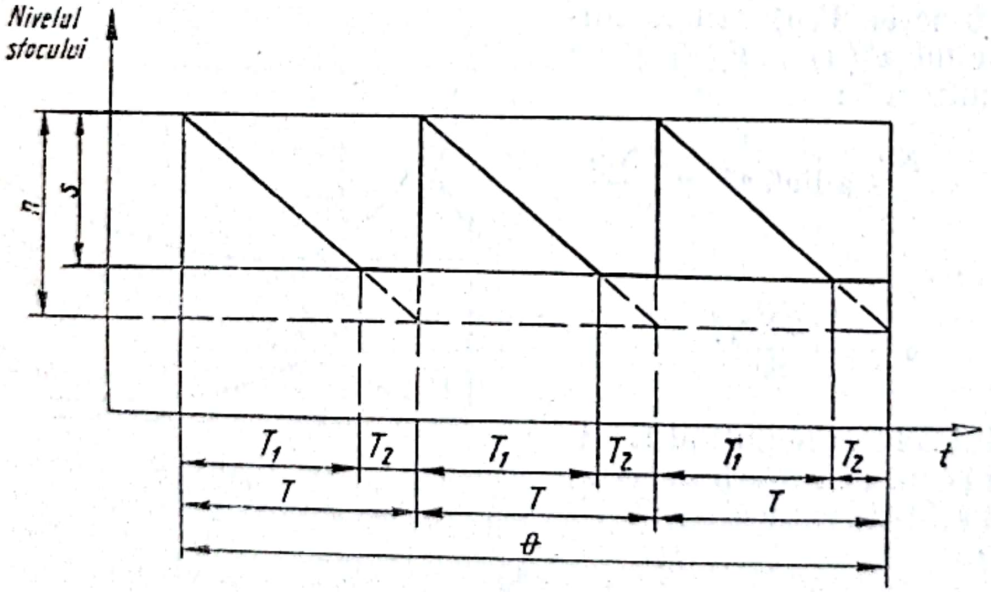


Fig. 5.8.

Cum

$$\frac{N}{n} = \frac{\theta}{T}$$

$$\frac{s}{n} = \frac{T_1}{T}; \quad \frac{T_2}{T} = \frac{n-s}{n}$$

Scanned with OKEN Scanner

rezultă că putem scrie :

$$\mathbf{F}(n, s) = \left[\frac{1}{2} s^2 \frac{\mathrm{T}}{\mathrm{n}} c_s + c_1 + \frac{1}{2}(n - s)^2 \frac{\mathrm{T}}{\mathrm{n}} c_p\right] \frac{\mathrm{N}}{n} =$$

$$= \frac{1}{2}\left[\frac{s^2 \theta c_s}{n} + \frac{\mathrm{N}}{n} c_l + \frac{(n - s)^2 \theta c_p}{2n}\right].$$

Rezultă că minimul funcţiei $\mathbf{F}(n, s)$ este atins pentru valorile n_0 şi s_0, soluţiile sistemului de ecuaţii : *

$$\begin{cases} -\dfrac{s^2 \theta c_s}{2n^2} - \dfrac{\mathrm{N} c_1}{n^2} + \dfrac{\theta c_p}{2} \dfrac{2n(n-s) - (n-s)^2}{n^2} = 0 \\ \dfrac{s \theta c_s}{n} - \dfrac{2(n-s)\theta c_p}{2n} = 0 \end{cases}$$

sau

$$\begin{cases} \theta c_p n^2 = 2\mathrm{N} c_1 + s^2 \theta (c_s + c_p) \\ s\theta (c_s + c_p) = n \theta c_p \end{cases}$$

de unde rezulta că

$$s = n \frac{c_p}{c_s + c_p}$$

$$\theta c_p n^2 \left(1 - \frac{c_p}{c_s + c_p}\right) = 2\mathrm{N} c_l.$$

(Cantitatea $\rho = \dfrac{c_p}{c_s + c_p}$ poartă numele de intensitate de ruptura de stoc sau factor de indisponibilitate.)
adică :

$$n^2 = \frac{2\mathrm{N} c_l}{\theta c_s} \cdot \frac{c_s + c_p}{c_q}$$

şi deci :

$$n_0 = \sqrt{2 \cdot \frac{\mathrm{N} c_l}{\theta c_s}} \cdot \sqrt{\frac{c_s + c_p}{c_p}}$$

---

* S-a presupus că cererea este continuă.

Scanned with OKEN Scanner

iar de aici rezultă că:

$$s_0 = n_0 \frac{c_p}{c_s + c_p} = \sqrt{2\frac{Nc_1}{\theta c_s}} \cdot \sqrt{\frac{c_p}{c_s + c_p}}.$$

Mai departe rezultă mărimea perioadei optime:

$$T_0 = n_0 \frac{\theta}{N} = \frac{\theta}{N} \cdot \sqrt{2\frac{Nc_1}{\theta c_s}} \cdot \sqrt{\frac{c_s + c_p}{c_p}} = \sqrt{2\frac{\theta c_1}{Nc_s}} \cdot \sqrt{\frac{c_s + c_s}{c_p}}$$

și

$$F_0 = \frac{1}{2} \cdot \frac{s_0^2 \theta c_0^s}{n_0} + \frac{Nc_1}{n_0} + \frac{(n_0 - s_0)^2 \theta c_p}{2n_0} = \sqrt{2N\theta c_s c_1} \cdot \sqrt{\frac{c_p}{c_s + c_p}}.$$

Exemplul 5.1. Să presupunem că cererea anuală pentru un material de construcții de masă este de 300 000 de unități. După natura materialului poate fi vorba de metri cubi (balast), bucăți (cărămizi) etc. În tot timpul anului, cererea este continuă și constantă pe intervale egale. Costul stocajului s-a evaluat prin calcule avînd în vedere gestiunea anterioară și s-a stabilit că este 0,006 lei pe unitate și pe zi. Cheltuielile de lansare a comenzii (cheltuieli administrative, plata achizitorului, delegați pentru recepție etc.) se ridică la 1 440 lei/lot și, desigur, sînt independente de volumul lotului. Se cere stabilirea lotului optim.

Cantitatea optimă de aprovizionat (lotul optim) va fi:

$$n_0 = \sqrt{\frac{2Nc_l}{\theta c_s}} = \sqrt{\frac{2 \times 300\,000 \times 1\,440}{360 \times 0{,}006}} = 20\,000 \text{ unități.}$$

Perioada la care trebuie să soseacă acest lot optim este dată de expresia:

$$T_0 = \frac{\theta}{N} n_0 = \frac{360 \times 20\,000}{300\,000} = 24 \text{ zile.}$$

Costul total al gestiunii este:

$$F_0 = \sqrt{2N\theta c_l c_s} = \sqrt{2 \times 300\,000 \times 360 \times 1\,440 \times 0{,}006} = 43\,200 \text{ lei.}$$

În acest exemplu, datele inițiale ale problemei (N, c_l, c_s) pot fi considerate ca rezultînd din studiul unui caz concret, de pildă, studiul gestiunii cimentului la un atelier de prefabricație. Trebuie să subliniem că obținerea acestor date inițiale necesită o prelucrare atentă și uneori laborioasă a numeroase observații și acte financiar-contabile.

Exemplul 5.2. Vom rezolva problema din exemplul anterior admițînd posibilitatea rupturii stocului și introducînd un cost de penalizare de 0,014 lei pe unitate și pe zi.

Scanned with OKEN Scanner

Factorul de indisponibilitate este

$$\rho = \frac{0{,}014}{0{,}006 + 0{,}014} = 0{,}7.$$

Aplicînd formulele prezentate în paragraful 5.3.2 obţinem: lotul necesar pentru evitarea ruperii stocului (n_0), stocul optim (s_0), lungimea optimă a perioadei de gestiune (T_0) şi costul total optim al gestiunii (C_0).

$$n_0 = 20\,000 \times \sqrt{\frac{1}{0{,}7}} = 24.000 \text{ unităţi};$$

$$s_0 = 24\,000 \times 0{,}7 = 16\,800 \text{ unităţi};$$

$$T_0 = 24 \times \sqrt{\frac{1}{0{,}7}} = 29 \text{ zile};$$

$$F_0 = 43\,200 \times \sqrt{0{,}7} = 36\,150 \text{ lei}.$$

Acest exemplu arată că în cazul unui cost de penalizare mic putem admite ruptura stocului în vederea obţinerii unor economii. O ipoteză implicită în acest caz este aceea că cererile nesatisfăcute într-o perioadă sînt amînate şi satisfăcute în perioada următoare.

5.3.3. Model de gestiune deterministă pentru un singur produs aplicabil unui ciclu de producţie pe o perioadă dată

Să presupunem că într-o perioadă de timp T trebuie satisfăcută cererea U de produse cu o intensitate constantă $u = \frac{U}{T}$. Presupunem că ritmul producţiei r este de asemenea constant şi că $r > u$, iar comenzile care nu au fost satisfăcute rămîn în evidenţă pentru a fi satisfăcute ulterior.

Să notăm cu

c_s costul de stocare a unei unităţi de produs în unitate de timp;

c_p costul de penalizare pe unitate de produs şi de timp;

c_l cheltuielile de lansare (cheltuielile unui ciclu de producţie);

q cantitatea de produse realizată într-un ciclu de producţie;

F cheltuielile medii pe unitatea de timp.

Atunci ca urmare a activităţii de producţie şi consumului produselor fabricate, pornind de la un moment dat, o perioadă de timp T_1 se produce şi se consumă produsul considerat. Dat fiind că ritmul producţiei este superior intensităţii consumului după perioada de timp T_1 se realizează un stoc S. Producţia încetează şi consumul este satisfăcut o perioadă de timp T_2 din acest stoc cînd se epuizează.

Scanned with OKEN Scanner

Începe o perioadă de timp T_3 cînd se constată o penurie de produse după care se reia producţia şi în perioada de timp T_4 se reuşeşte să se satisfacă cererile înregistrate anterior cît şi cele prezente după care începem să realizăm din nou stocul S deci, începem un nou ciclu de producţie şi consum.

Schematic această situaţie apare ca în figura 5.9.

Aşadar la început stocul este zero după care creşte pe măsură ce se produce pînă atinge nivelul S (perioada de timp T_1) apoi încetează producţia şi ca urmare a consumului nivelul stocului scade pînă atinge valoarea zero (perioada de timp T_2); mai departe comenzile de produs nu mai pot fi satisfăcute şi se înregistrează o penurie (perioada de timp T_3) pînă se acumulează deficitul s, după care începe iarăşi producţia, restanţele sînt eliminate treptat astfel că după o perioadă de timp T_4 ele au fost complet eliminate şi încep să apară din nou acumulări. Deci după o perioadă de timp $T_1 + T_2 + T_3 + T_4$ se reia din nou ciclul de producţie.

De-a lungul perioadei de timp $T_1 + T_2 + T_3 + T_4$ avem următoarele cheltuieli :

Cheltuieli de stocare de-a lungul perioadei de timp $T_1 + T_2$ şi ele sînt :

$$\frac{c_s S (T_1 + T_2)}{2}.$$

Cheltuielile de penalizare, de-a lungul perioadei de timp $T_3 + T_4$ şi ele sînt

$$\frac{c_p S (T_3 + T_4)}{2}.$$

Cheltuielile de lansare — cheltuieli fixe pentru întreaga perioadă, c_l.

Urmează că cheltuielile medii pe unitatea de timp sînt :

$$F(s, S, T_1, T_2, T_3, T_4) = \frac{\frac{1}{2} c_s S (T_1 + T_2) + c_p s (T_3 + T_4) + c_l}{T_1 + T_2 + T_3 + T_4}.$$

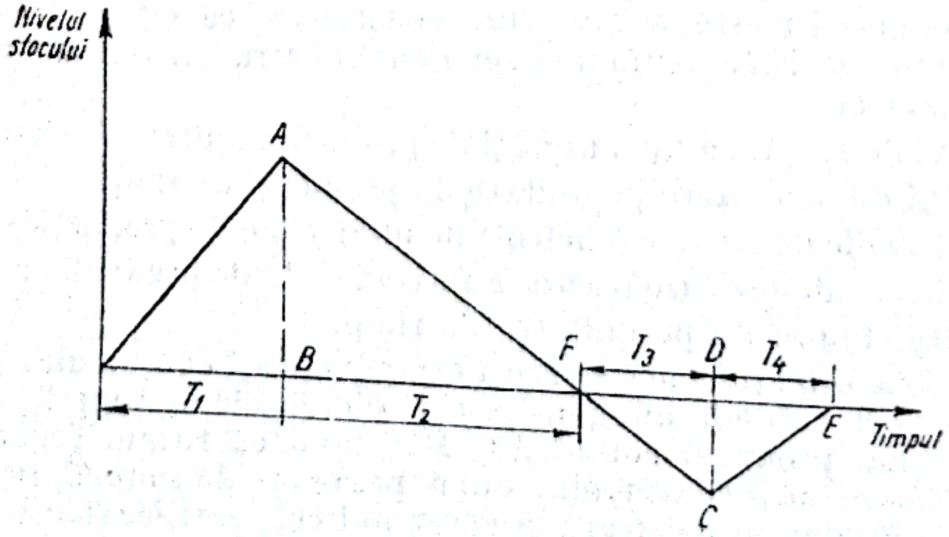


Fig. 5.9

Scanned with OKEN Scanner

Funcția cheltuieli medii pe unitatea de timp depinde la o primă vedere de șase variabile, dar nu toate variabilele sînt independente.

Într-adevăr în perioada de timp T_1 se produce cantitatea rT_1 însă din această cantitate se consumă uT_1 produse, așa că stocul maxim S este egal cu

$$S = T_1 (r - u).$$

Cum în perioada T_2 se consumă toate produsele stocate rezultă că $S = T_2 u$ și deci putem scrie

$$T_1 = \frac{S}{r-u} = \frac{T_2 u}{r-u}.$$

În perioada T_3 se acumulează restanțele cu intensitatea u, așa că $s = T_3 u$, iar în perioada T_4 sînt lichidate toate restanțele așa că $s = T_4(r-u)$ și deci aici

$$T_4 = \frac{s}{r-u} = \frac{T_3 u}{r-u}.$$

Înlocuind S, T_1, s, T_4 în $F(s, S, T_1, T_2, T_3, T_4)$ obținem :

$$F(T_2, T_3) = \frac{\frac{1}{2} ur(c_s T_2^2 + c_p T_3^2) + c_1(r-u)}{r(T_2 + T_3)}.$$

Pentru a pune în evidență cantitatea de produse realizată în intervalul de timp ce constituie ciclul de producție observăm că în condițiile menționate, cantitatea produsă este egală cu cantitatea consumată și de aici

$$q = u(T_1 + T_2 + T_3 + T_4) = \frac{ur(T_2+T_3)}{r-u}.$$

Să determinăm minimul funcției cheltuielilor medii pe unitatea de timp Pentru aceasta determinăm $T_2^{(0)}$ și $T_3^{(0)}$ din sistemul

$$\frac{\partial F(T_2, T_3)}{\partial T_2} = 0$$

$$\frac{\partial F(T_2, T_3)}{\partial T_3} = 0$$

adică :

$$\begin{cases} ur^2 c_s T_2 (T_2 + T_3) - r\left[\frac{1}{2} ur (c_s T_2^2 + c_p T_3^2) + c_1 (r-u)\right] = 0 \\ ur^2 c_p T_3 (T_2 + T_3) - r\left[\frac{1}{2} ur (c_s T_2^2 + c_p T_3^2) + c_1 (r-u)\right] = 0. \end{cases}$$

Scanned with OKEN Scanner

Rezolvind obținem perioadele optime

$$T_2^{(0)} \text{ și } T_3^{(0)}.$$

$$T_2^{(0)} = \sqrt{\frac{2c_p c_l (r-u)}{ur\, c_s (c_s + c_p)}} = \sqrt{\frac{2c_p c_l (1-u/r)}{uc_s(c_s+c_p)}}$$

și

$$T_3^{(0)} = \sqrt{\frac{2c_s c_l (r-u)}{ur\, c_p (c_s + c_p)}} = \sqrt{\frac{2c_s c_l (1-u/r)}{uc_p(c_s+c_p)}}.$$

De aici rezultă: lotul optim

$$q^{(0)} = \sqrt{\frac{2uc_l}{c_s} \frac{1}{1-u/r} \frac{c_s+c_p}{c_p}}$$

stocul optim

$$S^{(0)} = \sqrt{\frac{2uc_p c_l (1-u/r)}{c_s(c_s+c_p)}}$$

și valoarea minimă a cheltuielilor medii pe unitatea de timp:

$$F^{(0)} = \left(\frac{2u c_s c_p c_l (1-u/r)}{c_s + c_p}\right)^{1/2}.$$

Să facem acum anumite ipoteze suplimentare care ne vor conduce la unele probleme particulare de gestiune.

1) Să presupunem că ritmul producției r este foarte mare față de intensitatea cererii. În acest caz timpul total de producție $T_1 + T_4$ poate fi neglijat ceea ce este echivalent cu $\frac{r-u}{r} \simeq 1$, adică este vorba de o „producție instantanee".

Cu aceste precizări, cheltuielile medii pe unitatea de timp devin

$$F(T_2, T_3) = \frac{\frac{1}{2} u(c_s T_2^2 + c_p T_3^2) + c_l}{T_2 + T_3}$$

iar $T_2^{(0)}$, $T_3^{(0)}$, $q^{(0)}$, $S^{(0)}$, $F^{(0)}$ sînt respectiv

$$T_2^{(0)} = \sqrt{\frac{2c_p c_l}{uc_s(c_s+c_p)}}; \quad T_3^{(0)} = \sqrt{\frac{2c_s c_l}{uc_p(c_s+c_p)}}$$

$$q^{(0)} = \sqrt{\frac{2uc_l}{c_s} \frac{c_s+c_p}{c_p}}; \quad S^{(0)} = \sqrt{\frac{2uc_p c_l}{c_s(c_s+c_p)}}; \quad s^0 = \sqrt{\frac{2u c_s c_l}{c_p(c_s+c_p)}}$$

$$F^{(0)} = \sqrt{\frac{2u c_s c_p c_l}{c_s + c_p}}.$$

Scanned with OKEN Scanner

2) O altă ipoteză simplificatoare se obține atunci cînd nu se admite penuria de produse. Acest lucru este echivalent cu faptul că cheltuielile de penalizare sînt oricît de mari, adică putem face $c_p \to \infty$ în care caz $T_3 \to 0$.

În acest caz obținem următoarele valori pentru $T_2^{(0)}$, $q^{(0)}$, $S^{(0)}$ și $F^{(0)}$:

$$T_2^{(0)} = \sqrt{\frac{2c_l}{uc_s}}\ ;\ q^{(0)} = \sqrt{\frac{2uc_l}{c_s}}\ ;\ {}^{(0)} = 0\ ;\ F^{(0)} = \sqrt{2uc_sc_l}\ .$$

Aceste rezultate sînt identice cu cele obținute la 5.3.1 datorită utilizării unor ipoteze simplificatoare și faptului că intensitatea cererii este constantă.

5.3.4. Model de gestiune cu prețuri de achiziție sau cu cheltuieli de producție variabile

În modelul determinist general pe care l-am abordat anterior cu excepția cheltuielilor de lansare (presupuse fixe) cheltuielile de producție erau ignorate. Acest lucru este valabil dacă cheltuielile de producție pe unitatea de produs nu variază cu volumul producției iar cererea este satisfăcută în întregime (sau în modele de aprovizionare cheltuielile de aprovizionare pe unitatea de produs nu variază cu volumul comenzii).

În general însă cheltuielile de producție sînt funcție de volumul producției notat q și anume o funcție nedescrescătoare $f(q)$ cu $f(o) = 0$. Pentru cheltuieli de lansare $c_l \neq 0$ funcția $f(q)$ are un salt egal cu c_l. Uneori funcția $f(q)$ are și alte salturi care trebuie luate în considerație cînd se determină cantitatea optimă q ce trebuie achiziționată (produsă).

Să presupunem acum că intensitatea cererii de produse este U și să notăm ca și pînă acum cu c_s cheltuielile de stocare pe unitatea de produs în unitatea de timp iar cu c_l cheltuielile de lansare (fixe) a unei comenzi. Fie p prețul unitar al produsului cînd volumul comenzii este mai mic decît o cantitate Q și p' cînd volumul comenzii este mai mare sau egal cu Q, cu $p' < p$.

Atunci $f(q)$ are expresia

$$f(q) = \begin{cases} 0 & q = 0 \\ c_l + qp & 0 < q < Q \\ c_l + qp' & q \geqslant Q. \end{cases}$$

Dacă presupunem că nu se admite penurie de articole și că aprovizionarea se face instantaneu, atunci cheltuielile medii pe unitatea de timp se obțin pe baza relației $F(T_2) = \frac{1}{2}\ uc_s\ T_2 + \frac{c_l}{T_2}$ înlocuindu-se $T_2 = \frac{q}{u}$

și obținem

$$F(q) = \begin{cases} \frac{c_l u}{q} + pu + \frac{1}{2} c_s q \text{ dacă } o < q < Q \\ \frac{c_l u}{q} + p'u + \frac{1}{2} c_s q \text{ dacă } q \geqslant Q. \end{cases}$$

Funcția $F(q)$ are un punct de discontinuitate în Q și minimul ei se atinge fie în Q fie în q, rădăcină a ecuației $\frac{dF}{dq} = 0$.

Cum $$\frac{dF}{dq} = -\frac{c_l u}{q^2} + \frac{c_s}{2} \text{ dacă } q \neq Q$$

rezultă

$$q^{(0)} = \sqrt{\frac{2c_l u}{c_s}}.$$

(Minimul poate fi atins eventual și în punctul $q = Q$ care este punct de discontinuitate al funcției F.)

Să analizăm acum ce se întîmplă pentru $q^{(0)} \geqslant Q$ și pentru $q^{(0)} < Q$. Dacă $Q \leqslant q^{(0)}$ minimul funcției F se realizează pentru $q = q^{(0)}$ și obținem

$$F^{(0)} = p'u + \sqrt{2c_s c_l u}.$$

Dacă $Q > q^{(0)}$ și se achiziționează $q^{(0)}$ unități de produs, cheltuielile vor fi $pu + \sqrt{2c_s c_l u}$.

Pe de altă parte cheltuielile, dacă se decide să se achiziționeze Q unități de produs, la un preț mai scăzut, vor fi, $p'u + \frac{c_l u}{Q} + \frac{1}{2} c_s Q$.

Din aceste considerații rezultă că se decide asupra achiziționării a Q unități pe produs dacă:

$$p - p' \geqslant \frac{l}{u} \left[\frac{c_l u}{Q} + \frac{1}{2} c_s Q - \sqrt{2c_s c_l u} \right].$$

Desigur raționamentul utilizat se poate extinde și în cazul cînd există mai multe modificări ale prețului, funcția F avînd în acest caz mai multe puncte de discontinuitate și analiza urmează să fie făcută în funcție de aceste puncte.

Să analizăm însă o situație analoagă celei descrise și care apare în procesul de producție.

Scanned with OKEN Scanner

Desigur producerea unei cantităţi mai mari de produse necesită un timp de lucru suplimentar şi ca atare cheltuielile unitare de producţie ar putea creşte. Să presupunem că atunci cînd volumul produselor realizate $q \leqslant Q$ fiecare unitate de produs costă p unităţi iar dacă $q > Q$ costul unei unităţi de produs este $p' > p$.

În acest caz cheltuielile de producţie $f(q)$ vor fi :

$$f(q) = \begin{cases} 0 & q = 0 \\ c_l + pq & 0 < q < Q \\ c_l + pQ + p'(q - Q) & q \geqslant Q \end{cases}$$

Punînd pentru perioada de timp cît există stoc $T_2 = \frac{q}{u}$ şi exprimînd cheltuielile totale pe unitatea de timp prin

$$\frac{1}{2} u c_s T + \frac{c_l}{T_2}$$

rezultă că cheltuielile totale în unitatea de timp, în cazul nostru, vor fi

$$F(q) = \begin{cases} c_l \frac{u}{q} + pu + \frac{1}{2} c_s q & 0 < q < Q \\ \frac{c_l - (p' - p)Q u}{q} + p'u + \frac{1}{2} c_s q & q \geqslant Q \end{cases}$$

Se constată că funcţia $F(q)$ este continuă pentru $q > 0$ şi derivabilă în orice punct $q > 0$, $q \neq Q$. Pentru o astfel de funcţie minimul se realizează fie în punctul $q = Q$ fie în punctul în care se anulează derivata $F'(q)$.

Din această condiţie rezultă $q^{(0)}$ pentru care se obţin cheltuielile minime, $\min F(q) = F(q^{(0)})$.

5.3.5. Model determinist de stocaj pentru mai multe produse

În activitatea unei unităţi cel mai frecvent intervine cazul cînd stocul este evaluat simultan pentru mai multe produse.

Putem să rezolvăm problema gestiunii optime a stocului luînd în consideraţie separat fiecare produs.

Dar dacă capacităţile de depozitare sau posibilităţile de producţie sînt limitate nu mai putem considera problema pentru fiecare produs în parte. Atunci lucrurile se complică şi o astfel de problemă trebuie tratată separat. În cele ce urmează vom lua un caz simplu pe care-l vom trata cu ajutorul multiplicatorilor lui Lagrange. Să considerăm cazul a n produse, cînd producţia poate fi considerată instantanee şi cînd nu este permisă penuria de produse.

Scanned with OKEN Scanner

Să notăm cu u_i intensitatea cererii din produsul i, cu c_{li} cheltuielile de lansare și cu c_{si} cheltuielile de stocare, unde $i = 1,2,\ldots, n$.

Atunci înlocuind $T_{2i} = \frac{q_i}{u_i}$, cheltuielile totale pe unitatea de timp vor fi

$$\mathbf{F}(q_1, q_2,\ldots, q_n) = \sum_{i=1}^{n}\left[\frac{1}{2}c_{si}q_i + \frac{c_{li}u_i}{q_i}\right]$$

q_i reprezentînd cantitatea din produsul i ce intră în stoc.

Dacă nu impunem limitări asupra capacităților de stocare sau asupra producției, pentru determinarea cheltuielilor minime va trebui să rezolvăm sistemul de ecuații

$$\frac{\partial \mathbf{F}(q_1,\ldots,q_n)}{\partial q_i} = \frac{1}{2}c_{si} - \frac{c_{li}u_i}{q_i^2}$$

care conduce la valorile optimale :

$$q_i^{(0)} = \sqrt{\frac{2u_ic_{li}}{c_{si}}}$$

(Se arată că într-adevăr este vorba de un minim.)

Dacă există o restricție impusă de capacitatea de stocare atunci va trebui să luăm în considerație o problemă de minim cu restricții. Așa de exemplu dacă impunem ca în medie cantitatea stocată să nu depășească o anume valoare K, atunci $\mathbf{F}(q_1, q_2,\ldots, q_n)$ va trebui minimizată cu condiția

$$\frac{1}{2}\sum_{i=1}^{n} q_i \leqslant \mathrm{K}$$

și se determină cantitățile $q_i^{(0)}$ $(i = 1,2,\ldots, n)$.

Dacă $\sum_{i=1}^{n} q_i^{(0)} \leqslant 2\mathrm{K}$ nu se pun probleme; dar dacă $\sum_{i=1}^{n} q_i^{(0)} > 2\mathrm{K}$ atunci vor trebui reduse valorile găsite $q_i^{(0)}$.

Factorul *1/2* apare ca urmare a faptului că se ia în considerație stocul mediu, iar intensitățile u_i sînt constante.

Rezolvăm problema în modul următor :

Introducem

$$\mathbf{G}(q_1,\ldots, q_n;\ \lambda) = \sum_{i=1}^{n}\left[\frac{1}{2}c_{si}q_i + \frac{c_{li}u_i}{q_i}\right] + \lambda\left[\sum_{i=1}^{n} q_i - 2\mathrm{K}\right]$$

Scanned with OKEN Scanner

Din sistemul de ecuaţii

$$\frac{\partial G}{\partial q_i} = \frac{1}{2} c_{si} - \frac{c_{ti} u_i}{q_i^2} + \lambda = 0 \quad i = 1,2,\ldots,n$$

obţinem imediat :

$$q_i^{(0)} = \sqrt{\frac{2c_{ti} u_i}{c_{si} + 2\lambda}} \qquad i = 1,2,\ldots, \ n$$

şi determinăm pe λ astfel încît

$$\sum_{i=1}^{n} q_i^{(0)} = 2K$$

presupunînd deci că avem relaţia $\frac{1}{2} \sum_{i=1}^{n} q_i = K$ şi deci $F(q_1,\ldots, \ q_n) = G(q_1,\ldots, \ q_n)$.

Se poate proceda în mai multe moduri pentru aflarea lui λ dar cel mai simplu este ca λ să fie determinat cu aproximaţie pe bază de încercări. Mai constatăm că din relaţiile :

$$\frac{\partial G}{\partial q_i} = 0 \qquad i = 1,2,\ldots, \ n \text{ rezultă imediat.}$$

$$\frac{\partial F}{\partial q_i} = - \lambda \qquad i = 1,2,\ldots, \ n$$

şi de aici rezultă că $q_i^{(0)}$ se aleg astfel încît la majorarea cantităţilor q_i variaţia cheltuielilor pe unitatea de produs să fie aceeaşi oricare ar fi produsul $i = 1,2,\ldots, \ n$.

5.4. Model de gestiune cu perioada fixă şi cerere aleatoare cu posibilitatea de rupere a stocului

Pînă acum am presupus că cererea de produse este determinată cel puţin pe un interval de timp dat. În realitate apar însă foarte frecvent probleme pentru care această ipoteză nu mai este satisfăcută şi ca atare vom considera acum un model de gestiune cu perioadă fixă şi cerere aleatoare.

Scanned with OKEN Scanner

Datorită faptului că cererea este aleatoare se poate întîmpla ca stocul să se epuizeze în cursul unei perioade, aşa că vom lua în consideraţie şi această posibilitate. Aşadar, în timpul unei perioade T poate apărea una din situaţiile prezentate în figurile 5.10.

În figura 5.10, *a* se constată că cererea totală este inferioară stocului $s \geqslant u$, iar în figura 5.10, *b* cererea totală depăşeşte stocul $s \leqslant u$.

Vrem să determinăm cheltuielile medii de gestiune pe perioada T. Pentru aceasta vom presupune că se cunoaşte legea de repartiţie a variabilei

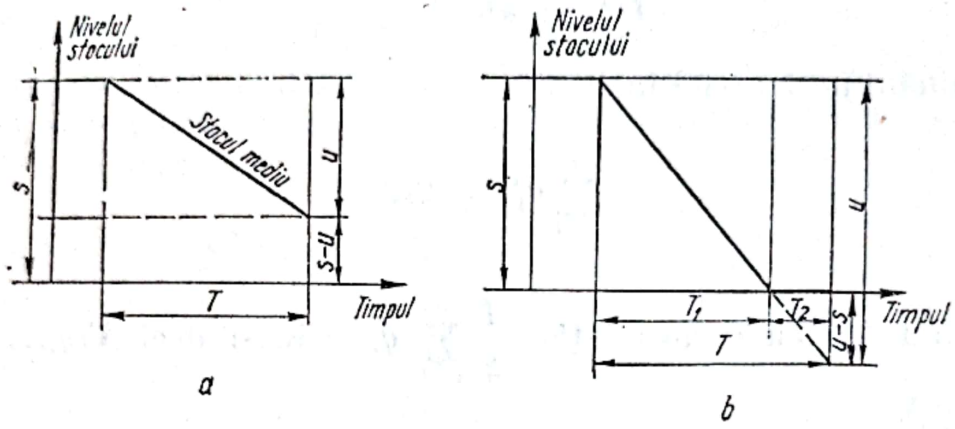


Fig. 5.10.

aleatoare U (cererea de produse); adică se cunosc probabilităţile $\mathbf{P}(U = u) = p(u)$, $u = 0,1, 2, \ldots$, într-un interval de timp de mărime T. Cheltuielile medii de gestiune vor fi compuse din cheltuielile medii de stocaj şi cheltuielile medii de penalizare (se neglijează cheltuielile de lansare prezentate în figurile 5.10.

Stocul mediu corespunzător situaţiei din figura 5.10, *a*) este dat de relaţia :

$$\frac{[s+(s-u)]T}{2T} = [s] - \frac{u}{2}$$

În situaţia din figura 5.10, *b* avem cheltuieli de stocaj numai pe intervalul de timp T_1. Stocul mediu este în acest caz $\frac{sT_1}{2T}$ Cum însă

$$\frac{T_1}{T} = \frac{s}{u}$$

rezultă stocul mediu : $\frac{s^2}{2u}$

Pe intervalul de timp T_2 se plătesc penalizări pentru cantitatea medie de articole lipsă care este dată de :

$$\frac{(u-s)\,T_2}{2T} = \frac{(u-s)^2}{2u}$$

De aici rezultă că cheltuielile medii de gestiune pe perioada T sînt date de:

$$\mathbf{F}(s) = c_s \sum_{u=0}^{s} \left(s - \frac{u}{2} \right) p(u) + c_s \sum_{u=s+1}^{\infty} \frac{s^2}{2u} p(u) + c_p \sum_{u=s+1}^{\infty} \frac{(u-s)^2}{2u} p(u)$$

(Dacă presupunem că **U** este variabilă aleatoare continuă se înlocuiesc sumele cu integrale.)

Nivelul optim al stocului este dat de acea valoare s_0 a lui s pentru care:

$$\mathbf{F}(s-1) > \mathbf{F}(s) < \mathbf{F}(s+1)$$

Să determinăm valoarea lui s_0 care satisface această dublă inegalitate

$$\mathbf{F}(s+1) = c_s \sum_{u=0}^{s+1} \left(s + 1 - \frac{u}{2} \right) p(u) + c_s \sum_{u=s+2}^{\infty} \frac{(s+1)^2}{2u} p(u) +$$

$$+ c_p \sum_{u=s+2}^{\infty} \frac{(u-s-1)^2}{2u} p(u) = \mathbf{F}(s) + (c_s + c_p) \left[P(U \leqslant s) + \right.$$

$$\left. + (s + 1/2) \sum_{u=s+1}^{\infty} \frac{p(u)}{u} \right] - c_p$$

Sau dacă notăm:

$$\mathbf{L}(s) = \mathbf{P}(\mathbf{U} \leqslant s) + (s + 1/2) \sum_{u=s+1}^{\infty} \frac{p(u)}{u}$$

condiţia $\mathbf{F}(s) < \mathbf{F}(s+1)$ revine la

$$c_p < (c_s + c_p)\, \mathbf{L}(s)$$

şi de aici

$$\rho = \frac{c_p}{c_s + c_p} < \mathbf{L}(s)$$

Procedînd analog cu $\mathbf{F}(s-1)$ putem scrie

$$\mathbf{F}(s-1) = c_s \sum_{u=0}^{s-1} \left(s - 1 - \frac{u}{2} \right) p\ (u) + c_s \sum_{u=s}^{\infty} \frac{(s-1)^2}{2u} p(u) +$$

$$+ c_p \sum_{u=s}^{\infty} \frac{(u-s+1)^2}{2u} p(u) = \mathbf{F}(s) - (c_s + c_p) \left[P(\mathbf{U} \leqslant s-1) + \right.$$

$$\left. + (s - 1/2) \sum_{u=s}^{\infty} \frac{p(u)}{u} \right] + c_p$$

Scanned with OKEN Scanner

iar inegalitatea $\mathbf{F}(s-1) > \mathbf{F}(s)$ devine

$$c_p > (c_s + c_p)\ \mathbf{L}(s-1)$$

sau

$$\rho = \frac{c_p}{c_s + c_p} > \mathbf{L}(s-1)$$

Așadar există o valoare s_0 astfel încît

$$\mathbf{L}(s_0 - 1) < \frac{c_p}{c_s + c_p} < \mathbf{L}(s_0)$$

sau scris explicit

$$\mathrm{P}(\mathrm{U} \leqslant s_0 - 1) + (s_0 - 1/2) \sum_{u=s_0}^{\infty} \frac{p(u)}{u} < \frac{c_p}{c_s + c_p} < \mathrm{P}(\mathrm{U} \leqslant s_0) +$$

$$+ (s_0 + 1/2) \sum_{u=s_0+1}^{\infty} \frac{p(u)}{u}$$

și de aici rezultă și regula practică de determinare a valorii s_0 care reprezintă stocul optim.

Exemplul 5.3. O întreprindere își propune să stabilească stocul optim de piese de schimb de un anumit tip pentru care s-au adunat următoarele informații.

Pe baza unor înregistrări statistice s-a stabilit că cererea de piese de schimb în unitatea de timp este o variabilă aleatoare cu următoarea repartiție:

Tabelul 5.1.

Numărul de piese de schimb solicitate	0	1	2	3	4	5	6
Probabilitatea	0,8	0,08	0,06	0,03	0,01	0,01	0,01

Pe baza unor studii economice s-a stabilit costul unitar de stocaj care este de *1* leu/piesă iar costul unitar de penalizare aferent lipsei de piese de schimb este de *200* lei pentru o piesă.

Se cere să se stabilească stocul optim de piese de schimb și cheltuielile medii totale corespunzătoare.

Soluție. Stocul optim s_0 este determinat de dubla inegalitate

$$\mathbf{F}(s_0 - 1) > \mathbf{F}(s_0) < \mathbf{F}(s_0 + 1)$$

Scanned with OKEN Scanner

care este echivalentă cu

$$P(U \leqslant s_0-1)+(s_0-1/2)\sum_{u=s_0}^{\infty}\frac{p(u)}{u} < \frac{c_p}{c_s+c_p} < P(U \leqslant s_0)+$$

$$+(s_0+1/2)\sum_{u=s_0+1}^{\infty}\frac{p(u)}{u}$$

Efectuăm calculele în tabelul 5.2.

Întrucît $\frac{c_p}{c_s+c_p}=\frac{200}{201}=0{,}9950248$

și

$$L(3)=0{,}99135<0{,}9950248<0{,}99620=L(4)$$

rezultă stocul optim $s_0=4$.

De asemenea rezultă cheltuielile medii pe unitatea de timp :

$$F(s)=c_s\sum_{u=0}^{s}(s-u/2)\,p(u)+c_s\sum_{u=s+1}^{\infty}\frac{s^L}{2u}\,p(u)+c_p\sum_{u=s+1}^{\infty}\frac{(u-1)^2}{2u}\,p(u)$$

și înlocuind $s=4$, $c_s=1$, $c_p=200$ obținem

$F(4)=4{,}845$ lei /unitatea de timp.

Tabelul 5.2.

U	S	$p(u)$	$\frac{p(u)}{u}$	$\sum_{u=s+1}^{\infty}\frac{p(u)}{u}$	$P(U \leqslant s)$	$L(s)=P(U<s)+(s+1/2)\sum_{u=s+1}^{\infty}\frac{p(u)}{u}$
0	0	0,80	—	0,1261	0,80	0,86305
1	1	0,08	0,08	0,0461	0,88	0,94915
2	2	0,06	0,03	0,0161	0,94	0,98025
3	3	0,03	0,01	0,0061	0,97	0,99135
4	4	0,01	0,0025	0,0036	0,98	0,99620
5	5	0,01	0,0020	0,0016	0,99	0,99880
6	6	0,01	0,0016	0,0000	1,00	1,00000

Exemplul 5.4. O întreprindere care produce echipamente complexe se angajează să livreze o anumită piesă de schimb beneficiarilor pînă la casarea echipamentului și să asigure totodată depanarea provocată de această piesă.

Cererea de piese de schimb din partea beneficiarilor este aleatoare și dată fiind importanța piesei de schimb, întreprinderea produce un a numit număr odată cu executarea echipamentului.

Scanned with OKEN Scanner

Cunoscîndu-se că preţul unei piese este de c unităţi monetare şi că lipsa unei astfel de piese pentru depanarea echipamentului se ridică la c' unităţi monetare (daunele pretinse de beneficiar, cheltuielile pentru asigurarea depanării etc.) se cere să se stabilească stocul minim de piese de schimb ce trebuie asigurat ştiind că cererea aleatoare **U** are densitatea de repartiţie $f(u)$. (Se presupune că cheltuielile de stocaj sînt neglijabile.)

Soluţie. Să exprimăm cheltuielile medii pe o perioadă de timp T. Din datele problemei rezultă că aceste cheltuieli medii sînt date de

$$\mathbf{F}(s) = c\mathrm{T}\int_0^s \left(s - \frac{u}{2}\right) f(u)\,du + c\mathrm{T}\int_s^\infty \frac{s^2}{2u} f(u)\,du +$$

$$+ c'\mathrm{T}\int_s^\infty \frac{(u-s)^2}{2u} f(u)\,du$$

Pentru determinarea stocului optim rezolvăm ecuaţia

$$\frac{\mathrm{d}\mathbf{F}(s)}{\mathrm{d}s} = 0$$

Cum

$$\frac{\mathrm{d}\mathbf{F}(s)}{\mathrm{d}s} = c\mathrm{T}\int_0^s f(u)\,du + c\mathrm{T}\int_s^\infty \frac{s}{u} f(u)\,du + c'\mathrm{T}\int_s^\infty \frac{s-u}{u} f(u)\,du =$$

$$= \mathrm{T}(c + c')\left[\int_0^s f(u)\,du + s\int_s^\infty \frac{f(u)}{u}\,du\right] - c'\mathrm{T}$$

rezultă că s este soluţia ecuaţiei

$$s\int_s^\infty \frac{f(u)}{u}\,du + \int_0^\infty f(u)\,du = \frac{c'}{c+c'}$$

În cazul particular cînd densitatea de repartiţie este $f(u) = ue^{-u}$, $u > 0$.

obţinem $\int_s^\infty \frac{f(u)}{u}\,du = e^{-s}$; $\int_0^s f(u)du = -e^{-s}(s+1) + 1$

şi valoarea stocului rezultă din relaţia

$$1 - e^{-s_0} = \frac{c'}{c+c'}$$

sau

$$s_0 = \ln\frac{c+c'}{c}$$

Scanned with OKEN Scanner

Adesea se întîmplă că cheltuielile de stocaj sînt neglijabile dar intervin pierderi inevitabile pentru produsele nevîndute. În aceste pierderi intră de exemplu cheltuielile pentru producerea respectivului produs, pierderi rezultate din blocarea fondurilor care urmau să fie rulate etc.

O astfel de situaţie poate interveni în industrie după cum va rezulta din exemplul care urmează :

Exemplul 5.5. Să presupunem că o întreprindere lansează pe piaţă un tip îmbunătăţit dintre produsele sale şi că seria este limitată la un anumit număr de exemplare. Odată cu lansarea pe piaţă a acestui tip de produs se pune problema determinării numărului de piese de schimb de un anumit fel, care trebuie stocate.

Acest număr de piese de schimb nu trebuie să fie nici exagerat de mare întrucît excesul antrenează pierderi mari ca urmare a imobilizării de fonduri şi nici prea mic deoarece lipsa pieselor de schimb înseamnă obligaţia întreprinderii de a le executa chiar dacă nu se mai fabrică tipul de produs, ceea ce revine la un preţ mult mai ridicat.

Să admitem că pierderea unitară la articolele nevîndute este de c_1 unităţi monetare, iar pierderile aduse de fabricarea suplimentară a articolelor lipsă din stoc, atunci cînd la o cerere se constată lipsa articolului, este de c_2 unităţi monetare.

Dintr-un studiu al cererii se apreciază că **U** are repartiţia :

$$\mathbf{U}: \begin{pmatrix} 0 & 1 & \dots & n & \dots \\ p(0) & p(1) & & p(n) & \dots \end{pmatrix}$$

Soluţie. Vom calcula valoarea medie a cheltuielilor efectuate. Întrucît cheltuielile de stocaj sînt neglijabile în raport cu c_1 şi c_2, rezultă că nu intervine perioada de gestiune T.

Dacă notăm cu s numărul de piese de schimb pe care le stocăm odată cu lansarea seriei de produse, atunci cheltuielile medii sînt date de expresia :

$$\mathbf{F}(s) = c_1 \sum_{u=0}^{s} (s-u)\, p(u) + c_2 \sum_{u=s+1}^{\infty} (u-s)\; p(u)$$

Vom determina valoarea lui s pentru care

$$\mathbf{F}(s-1) > \mathbf{F}(s) < \mathbf{F}(s+1)$$

Cum

$$\mathbf{F}(s+1) = \mathbf{F}(s) + (c_1 + c_2)\, \mathbf{P}(\mathbf{U} \leqslant s) - c_2$$

şi

$$\mathbf{F}(s-1) = \mathbf{F}(s) - (c_1 + c_2)\, \mathbf{P}(\mathbf{U} \leqslant s-1) + c_2$$

rezultă că valoarea s_0 a lui s care minimizează cheltuielile $\mathbf{F}(s)$ verifică dubla inegalitate

$$\mathbf{P}(\mathbf{U} \leqslant s_0 - 1) < \frac{c_2}{c_1 + c_2} < \mathbf{P}(\mathbf{U} \leqslant s_0).$$

Scanned with OKEN Scanner

5.5. Metode euristice de gestiune a stocurilor

În multe întreprinderi, numărul articolelor care intră şi ies din magazie este foarte mare. Din această cauză nu este practic posibil să se urmărească fenomenul de stocare pentru toate articolele. Pentru a stăpîni totuşi procesul în mare, s-a imaginat o metodă de control selectiv numită metoda ABC.

Folosirea metodei ABC în întreprinderi constă în alegerea unui criteriu corespunzător pe baza căruia se efectuează împărţirea materialelor în trei grupe ABC. Un asemenea criteriu poate fi valoarea materialului, modul de folosire a acestuia în producţie sau provenienţa lui din import.

Controlul selectiv al stocurilor

Fiecare articol are un anumit consum anual şi corespunzător o valoare. Ordonînd articolele crescător după consumul anual şi descrescător după valoare, putem împărţi articolele din stocuri în trei grupe :

— grupa A, în care intră articolele scumpe care reprezintă cantitativ 10% din stoc şi 70% valoric ;

— grupa B conţine articole care reprezintă 20% cantitativ şi 20% valoric ;

— grupa C, articole care reprezintă cantitativ 70% şi valoric 10%.

Pentru că în grupa A sînt puţine articole, se poate controla zilnic nivelul stocurilor, înregistrîndu-se variaţia cererii şi totodată se supraveghează de aproape respectarea termenelor de către furnizori. Deci se înlocuieşte o parte din stocul tampon de articole scumpe printr-un control al gestiunii mai strîns. Această decizie este eficientă pentru că duce la o reducere apreciabilă a fondurilor blocate în stocuri.

La articolele din grupa C care sînt mai ieftine este mai eficient să se suporte costuri de stocare, decît să se plătească personalul care ar fi indispensabil, pentru sporirea controlului.

Pentru articolele din grupa B baza controlului rămîne stocul tampon.

Metoda ABC permite să se practice o politică de reaprovizionare diferită în cele trei grupe, cu un control mai intens asupra articolelor din grupa A.

Metodele de gestiune bazate pe utilizarea modelelor matematice presupun o mare exactitate în determinarea costurilor care intră în calcule. Din nefericire, adeseori acest lucru nu se poate realiza, gestiunea stocurilor reducîndu-se la simpla evidenţă a mişcării resurselor şi aplicarea unei politici de reaprovizionare bazate pe intuiţie şi experienţă.

Trebuie arătat, de asemenea, că, în practică, gestiunea stocurilor este determinată de existenţa unor normative privind limitarea imobilizării fondurilor circulante.

Considerăm însă că avantajele pe care le-ar aduce aplicarea unor metode ştiinţifice în politica stocurilor ar justifica cheltuielile necesare pentru o evidenţă mai analitică a costurilor, precum şi eforturile de a reexamina şi recalcula normativele privind imobilizarea fondurilor circulante.

În bibliotecile de programe ale calculatoarelor există pachete importante de programe care permit îmbunătăţirea gestiunii stocurilor (BOMP—IBM, ORACLE—FELIX etc.).

Se înţelege că aplicarea lor necesită efectuarea în prealabil a unei analize de sistem.

Scanned with OKEN Scanner

Capitolul VI

PROGRAMAREA DINAMICĂ

6.1. Introducere

În modelarea matematică a fenomenelor economice variabile în timp se ataşează fenomenului o funcţie al cărei optim trebuie căutat în condiţii bine stabilite pe baza fenomenului economic concret considerat.

Unul din instrumentele cele mai eficiente pentru căutarea optimului funcţiei ataşate fenomenului economic este relativ nou şi a fost pus în evidenţă de matematicianul american Richard Bellman. Acest instrument este cunoscut azi sub numele de *programare dinamică*. Pe scurt putem spune că programarea dinamică este o metodă de optimizare a sistemelor utilizînd reprezentări matematice ale lor, în care se operează pe faze sau secvenţe, sau aşa cum le vom spune şi noi mai departe, perioade. Astfel de sisteme se întîlnesc frecvent în studiile economice sau la realizarea unor programe privind tehnicile cele mai avansate cum ar fi de exemplu cele ce privesc navigaţia cosmică. Punctul de plecare al acestei metode îl constituie „teorema de optimalitate" şi care sub forma cea mai generală poate fi enunţată ca un principiu, „principiul de optimalitate".

Acest „principiu de optimalitate" este atît de simplu încît pare aproape evident după ce a fost bine înţeles. Importanţa lui şi eficienţa metodelor de optimizare secvenţială la care a dat naştere devin din ce în ce mai evidente pe măsură ce ne dăm seama de caracterul secvenţial al numeroaselor probleme economice.

Referitor la fenomenele economice este aproape evident caracterul secvenţial al problemelor privind îndeosebi aplicaţiile în microeconomie.

Cînd este vorba de regularizarea producţiei şi a stocurilor de materiale, de gestiune a echipamentelor, de prospecţiunile miniere, de investiţii, fără a mai vorbi de probleme macroeconomice cum ar fi cea a planificării naţionale, caracterul secvenţial al problemelor justifică utilizarea unor metode adecvate care să permită pe lîngă efectuarea calculelor de optimizare şi clarificarea problemelor în introducerea unor concepte precise cum ar fi : criteriul de decizie, politica sau strategia de urmat, influenţa informaţiei asupra calităţii deciziilor. Datorită importanţei acestei noţiuni în studiul şi analiza fenomenelor economice, toate vor trebui definite cît mai precis posibil.

Scanned with OKEN Scanner

Problemele acestea pot fi considerate atît ca probleme secvențiale discrete în timp cît și în timp continuu.

Noi vom considera doar probleme secvențiale discrete în timp, pentru a ocoli o serie de dificultăți ce apar în acest din urmă caz.

Cînd variabilele sînt considerate continue, la trecerea de la o fază la cea următoare apar probleme delicate de topologie algebrică pentru raportarea valorilor optime. Unul din mijloacele eficiente pentru eliminarea acestor dificultăți constă în înlocuirea mulțimilor continue cu mulțimi discrete numărabile.

În multe cazuri însă procedînd în felul acesta se pierde din rigurozitatea discuției și analizei care sînt proprii calculului infinitesimal.

Cu toate acestea în diferite probleme practice se pot accepta ca satisfăcătoare mulțimile discrete.

O altă noțiune care intervine în studiul programelor dinamice este și cea de orizont economic (numărul de faze sau perioade pe care se desfășoară programul dinamic).

Noi vom considera aici doar problemele cu orizont economic limitat.

Noțiunea de orizont economic este deosebit de importantă pentru problemele de programare dinamică și ea conduce de cele mai multe ori la examinarea convergenței pe un orizont infinit. În multe cazuri reale din economie prin introducerea actualizării, ceea ce dealtfel se realizează, programele dinamice pot fi făcute convergente.

Vom distinge de asemenea programele dinamice și după natura viitorului în care se desfășoară aceste programe. Astfel, vom avea programe dinamice în viitor sigur și programe dinamice în viitor aleator.

Din mulțimea programelor dinamice ne vom opri în acest capitol doar la programe dinamice discrete în viitor sigur și cu orizontul limitat.

6.2. Modelul matematic al programelor dinamice discrete ce se desfășoară în viitor sigur și cu orizont limitat

Fie S un sistem a cărui stare la fiecare moment n, $n \in \{0, 1, 2, \ldots, N\}$ este definită cu ajutorul unei variabile de stare x_n. Această variabilă de stare este astfel definită încît să caracterizeze complet starea sistemului, adică referindu-ne la evoluția viitoare a sistemului să nu existe posibilitatea distingerii a două evoluții diferite ale sistemului pe parcursul perioadelor de la 1 la n și care să conducă la aceeași variabilă de stare x_n.

O perioadă n a programului dinamic se definește ca un cuplu de momente $(n-1,\ n)$.

În orice perioadă n, există o decizie ce trebuie luată referitor la valorile variabilei de stare, x_n. În general domeniul de variație al variabilelor x_n este limitat de restricții ce depind de valorile variabilei x_{n-1}.

Vom presupune (într-un mod foarte general) că restricțiile problemei definesc pentru fiecare valoare a momentului n și a variabilei de stare x_n o mulțime de valori posibile ale variabilei x_{n+1}, mulțime pe care o vom nota $\Gamma_{n+1}x_n$.

Scanned with OKEN Scanner

În felul acesta va trebui să avem :

$$x_{n+1} \in \Gamma_{n+1} x_n$$

Reciproc, dacă mulțimea $\Gamma_{n+1} x_n$ a valorilor variabilei de stare x_{n+1} este dată, atunci, va trebui să fie satisfăcută relația :

$$x_n \in \Gamma_{n+1}^{-1} x_{n+1}$$

unde : $\Gamma_{n+1}^{-1} x_{n+1}$ este mulțimea valorilor posibile ale lui x_n astfel încît $x_{n+1} \in \Gamma_{n+1} x_n$.

În termeni matematici Γ_{n+1} este o aplicație — neunivocă în general — a mulțimii valorilor lui x_n în mulțimea valorilor lui x_{n+1}, iar Γ_{n+1}^{-1} este aplicația inversă aplicației Γ_{n+1}.

Rezultă atunci că restricțiile referitoare la x_n se pot scrie sub forma condensată

$$x_n \in \Gamma_n x_{n-1} \cap \Gamma_{n+1}^{-1} x_{n+1}$$

Dacă acum se pleacă de la o stare inițială x_0 dată, domeniul evoluției posibile a sistemului la momentul n, notat $X_n(x_0,.)$ va putea fi definit din aproape în aproape cu ajutorul aplicațiilor Γ_n după cum urmează :

$$\mathbf{X}_0 = \{x_0\}$$

$$x_1 \in \Gamma_1 x_0, \qquad \mathbf{X}_1(x_0,.) = \Gamma_1 \mathbf{X}_0 = \Gamma_1 x_0$$

$$x_2 \in \Gamma_2 x_1, \qquad \mathbf{X}_2(x_0,.) = \Gamma_2 \mathbf{X}_1 = \bigcup_{x_1 \in \mathbf{X}_1} \Gamma_2 x_1$$

$$x_3 \in \Gamma_3 x_2, \qquad \mathbf{X}_3(x_0,.) = \Gamma_3 \mathbf{X}_2 = \bigcup_{x_2 \in \mathbf{X}_2} \Gamma_3 x_2$$

$$x_n \in \Gamma_n x_{n-1}; \qquad \mathbf{X}_n(x_0,.) = \Gamma_n \mathbf{X}_{n-1} = \bigcup_{x_{n-1} \in \mathbf{X}_{n-1}} \Gamma_n x_{n-1}.$$

Dacă se impune că după N perioade să se ajungă la o anumită stare x_N, atunci mulțimile de valori posibile indicate mai sus trebuie să devină restricții, utilizînd succesiv restricțiile $\Gamma_n^{-1} x_n$ pentru $n = N, N-1, N-2, \ldots$

Se obțin în acest fel mulțimile $\mathbf{X}_n(x_0, x_N)$ care verifică relația :

$$\mathbf{X}_n(x_0, x_N) = \Gamma_n \mathbf{X}_{n-1}(x_0, x_N) \cap \Gamma_{n+1}^{-1} \mathbf{X}_{n+1}(x_0, x_N) \quad n = 1, 2, \ldots, N-1.$$

unde $\mathbf{X}_0(x_0, x_N)$ și $\mathbf{X}_N(x_0, x_N)$ se reduc la mulțimi formate din cîte un singur element.

$$\mathbf{X}_0(x_0, x_N) = \{x_0\}; \mathbf{X}_N(x_0, x_N) = \{x_N\}$$

O noțiune fundamentală în studiul programelor dinamice este noțiunea de politică de urmat pe care o vom defini după cum urmează :

Se numește „politică de la x_0 la x_N" și se va nota $\pi_{0,N}(x_0, x_N)$ orice succesiune $(x_0, x_1, x_2, \ldots, x_{N-1}, x_N)$ astfel încît să fie îndeplinite relațiile :

$$\pi_{0,N}(x_0, x_N) = (x_0, x_1, \ldots, x_{N-1}, x_N)$$

$$x_n \in \Gamma_n x_{n-1} \cap \Gamma_{n+1}^{-1} x_{n+1}, \quad n = 1, 2, \ldots, N-1.$$

Scanned with OKEN Scanner

Problema pe care urmărim s-o rezolvăm este problema aflării politicii π^* optimale în raport cu un anumit criteriu pe care-l vom defini mai jos.

Mai întîi fiecărui cuplu (x_{n-1}, x_n) i se asociază o valoare determinată $v_n(x_{n-1}, x_n)$ numită „valoare imediată" corespunzătoare perioadei n.

Semnificația economică a acestei valori este următoarea: dacă la momentul $(n-1)$ sistemul se află în starea x_{n-1}, valoarea deciziei x_n este $v_n(x_{n-1}, x_n)$.

Avînd valorile imediate $v_n(x_{n-1}, x_n)$ pentru orice perioadă $n = 1, 2, \ldots, N$, oricărei politici π i se asociază o valoare totală $s(\pi)$ definită prin relația

$$s(\pi) = \sum_{n=1}^{N} v_n(x_{n-1}, x_n)$$

Criteriul de alegere pe care-l vom reprezenta va consta în minimizarea (sau maximizarea, în funcție de natura concretă a valorilor v_n) valorii $s(\pi)$ a politicii. Subliniem faptul că valoarea $s(\pi)$ este definită aditiv. Apar și situații cînd valoarea $s(\pi)$ este definită multiplicativ sau chiar ca produs de convoluție.

Dacă notăm cu Π mulțimea tuturor politicilor și dacă presupunem pentru a ne fixa ideile că dorim să maximizăm valoarea politicii, atunci politicile optimale π^* vor fi definite prin relația

$$s(\pi^*) = \max_{\pi \in \Pi} s(\pi)$$

Mai general putem scrie:

$$s(\pi^*) = \operatorname{opt}_{\pi \in \Pi} s(\pi)$$

înțelegînd prin „opt", în funcție de natura problemei unul dintre operatorii „max" sau „min".

Sîntem acum în măsură să enunțăm și să demonstrăm teorema de optimalitate. Pentru aceasta vom defini și noțiunea de subpolitică a unei politici date.

Definiție. Pentru orice $0 \leqslant i \leqslant j \leqslant N$, vom numi „subpolitică de la x_i la x_j" orice succesiune

$$\pi_{ij}(x_i, x_j) = (x_i, x_{i+1}, \ldots, x_n, \ldots, x_j)$$

astfel încît să fie satisfăcute relațiile

$$x_n \in \Gamma_n x_{n-1} \cap \Gamma_{n+1}^{-1} x_{n+1}, \quad n = i+1, \; i+2, \ldots, \; j-1$$

Orice subpolitică de la x_i la x_j va fi optimală de la x_i la x_j dacă valoarea sa:

$$s_{ij}(\pi_{ij}) = \sum_{n=i+1}^{j} v_n(x_{n-1}, x_n)$$

este maximă (sau minimă, în funcție de natura concretă a problemei) printre toate subpoliticile de la x_i la x_j.

Scanned with OKEN Scanner

Să dăm acum în fine

Teorema de optimalitate. Orice subpolitică $\pi_{ij}(x_i, x_j)$ extrasă dintr-o politică optimală $\pi^*_{0,N}$ (x_0, x_N) este ea însăşi optimală de la x_i la x_j.

Demonstraţie. Fie o politică optimală

$$\pi^* = \{x_0, x_1, \ldots, x_{i-1}, x_i^*, x_{i+1}^*, \ldots, x_j^*, x_{j+1}, \ldots, x_N\}$$

În acest caz subpolitica formată din partea $\{x_i^*, x_{i+1}^*, \ldots, x_j^*\}$ a lui π^* este optimală de la x_i^* la x_j^* deoarece în caz contrar ar exista o subpolitică de la x_i^* la x_j^* care ar avea o valoare mai mare (în cazul maximului) sau mai mică (în cazul minimului) ceea ce ar atrage după sine îmbunătăţirea politicii π^* înlocuind porţiunea $\{x_i^*, \ldots, x_j^*\}$ cu această nouă subpolitică.

Din această teoremă rezultă imediat următorul

Corolar. În mulţimea politicilor care conţin o subpolitică π_{0k} (x_0, x_{0k}) dată (optimală sau nu), cea mai bună este aceea care se obţine completînd subpolitica dată cu o subpolitică optimală

$$\pi^*_{k,N}(x_k, x_N)$$

Acest corolar ce constituie aşa-numitul „principiu de optimalitate" este datorat, ca şi teorema dealtfel, lui R. Bellman şi el stă la baza metodei programării dinamice.

El ne dă posibilitatea ca în urma considerării succesive a 1, 2, ..., N perioade să construim progresiv o politică optimală. Pentru aceasta se poate porni fie de la prima perioadă, fie de la ultima perioadă, fie de la o perioadă oarecare.

Să construim efectiv o politică optimală pornind de la perioada *1* către perioada N. Vom spune în cazul acesta că optimizarea se efectuează de la trecut spre viitor.

O să notăm cu $f_{0,N}(x_0, x_N) = s(\pi^*) = \operatorname{opt}_{\pi \in \Pi} s(\pi)$ şi cu $f_{0,n}(x_0, x_n)$ valoarea optimală a subpoliticii de la x_0 la x_n.

Cu aceste precizări procedăm în felul următor:

În perioada *1*, pentru orice decizie posibilă

$$x_1 \in \mathbf{X}_1(x_0, .) = \mathbf{\Gamma}_1 x_0$$

cunoaştem valoarea $v_1(x_0, x_1) = f_{0,1}(x_0, x_1)$.

Considerăm perioadele *1* şi *2* împreună.

Pentru orice decizie

$$x_2 \in \mathbf{X}_2(x_0, .) = \mathbf{\Gamma}_2 x_1$$

se caută mulţimea subpoliticilor de la x_0 la x_2.

Aceste subpolitici

$$\pi_{0,2}(x_0, x_2) = \{x_0, x_1, x_2\}$$

depind de valoarea x_1 aleasă, valoare care trebuie să satisfacă condiţia:

$$x_1 \in \mathbf{\Gamma}_1 x_2 \cap \mathbf{\Gamma}_2^{-1} x_2 = \mathbf{X}_1(x_0, x_2)$$

Valoarea unei astfel de subpolitici $\pi_{0,2}(x_0, x_2)$ este dată de

$$s_{0\ 2}(\pi_{0,2}) = v_1(x_0, x_1) + v_2(x_1, x_2)$$

Scanned with OKEN Scanner

Putem obține acum subpoliticile optimale de la x_0 la x_2 căutînd valoarea optimă a lui $s_{0,2}$:

$$f_{0,2}(x_0,\ x_2) = \underset{x_1 \in \mathbf{X}_1(x_0,\ x_2)}{\text{opt}} [v_1(x_0,\ x_1) + v_2(x_1,\ x_2)]$$

care ne dă posibilitatea să determinăm x_1^*.

Considerăm acum mulțimea perioadelor *1, 2, 3.*

Pentru orice:

$$x_3 \in \mathbf{X}_3(x_3, .) = \mathbf{\Gamma}_3 \mathbf{X}_2(x_0, .)$$

se determină:

$$\mathbf{X}_2(x_0,\ x_3) = \mathbf{X}_2(x_0, .) \cap \mathbf{\Gamma}_3^{-1} x_3$$

După teorema de optimalitate partea $(x_0,\ x_1,\ x_2)$ a unei subpoliticii optimale de la x_0 la x_3 este o subpolitică optimală de la x_0 la x_2. Pentru a afla subpoliticile optimale de la x_0 la x_3 este suficient să considerăm subpoliticile de la x_0 la x_3 care sînt optimale de la x_0 la x_2. Valoarea optimală de la x_0 la x_3 va fi atunci

$$f_{0,3}(x_0,\ x_3) = \underset{x_2 \in \mathbf{X}_2(x_0, x_3)}{\text{opt}} [f_{0,2}(x_0,\ x_2) + v_3(x_2,\ x_3)]$$

Subpoliticile optimale de la x_0 la x_3 vor fi acelea care trec prin valorile x_2^* pentru care este atinsă valoarea maximă (sau minimă).

Continuînd tot așa mai departe procedeul, pentru aflarea valorii $f_{0,n-1}(x_0,\ x_{n-1})$ pentru orice $x_{n-1} \in \mathbf{X}_{n-1}(x_0, .)$ se determină $\mathbf{X}_{n-1}(x_0, .) = \mathbf{\Gamma}_{n-1} \mathbf{X}_{n-2}(x_0, .)$

Considerăm perioadele 1, 2,..., n împreună.

Pentru orice $x_n \in \mathbf{X}_n(x_0, .)$ se determină

$$\mathbf{X}_{n-1}(x_0,\ x_n) = \mathbf{X}_{n-1}(x_0, .) \cap \mathbf{\Gamma}_n^{-1} x_n$$

și se calculează valoarea optimală de la x_0 la x_n:

$$f_{0,n}(x_0,\ x_n) = \underset{x_{n-1} \in \mathbf{X}_{n-1}(x_0,\ x_n)}{\text{opt}} [f_{0,n-1}(x_0,\ x_{n-1}) + v_n(x_{n-1},\ x_n)]$$

Dacă x_{n-1}^* este o valoare a lui x_{n-1} care optimizează membrul al doilea, orice subpolitică de la x_0 la x_n care se obține prin adăugarea lui x_n la o subpolitică optimală de la x_0 la x_{n-1}^* este, la rîndul ei, optimală de la x_0 la x_n:

$$\pi_{0,n}^*(x_0,\ x_n) = \{\pi_{0,n-1}^*(x_0,\ x_{n-1}^*),\ x_n\}$$

În acest mod se obțin subpoliticile optimale de la x_0 la x_n. În momentul cînd am ajuns la $n = \mathrm{N}$ s-a rezolvat problema găsirii politicii optimale pentru x_N fixat.

— În anumite probleme nu se impune valoarea x_N. Se va alege atunci acea politică optimală de la x_0 la x_N care are cea mai mare valoare (cînd se maximizează valoarea totală).

Este clar că în mod necesar va trebui să conducem optimizarea așa cum s-a făcut mai sus, adică în sensul valorilor crescătoare ale lui n.

S-ar putea întîmpla ca x_N să fie fixat iar x_0 să ia diferite valori. Ar trebui atunci pornit de la $n = \mathrm{N}$ și să avansăm cu optimizarea în sensul valorilor descrescătoare ale lui n.

Scanned with OKEN Scanner

Metoda este evident aceeaşi însă $f_{0,n}(x_0, x_n)$ şi $\pi^*_{0,n}(x_0, x_n)$ se vor înlocui prin $f_{n,N}(x_0, x_n)$ şi $\pi^*_{n,N}(x_n, x_N)$ respectiv.

Se obţin în acest caz ecuaţiile de recurenţă

$$f_{n,N}(x_n, x_N) = \operatorname*{opt}_{x_{n+1} \in X_{n+1}(x_n, x_N)} [v_{n+1}(x_n, x_{n+1}) + f_{n+1,N}(x_{n+1}, x_N)]$$

unde $n = N - 2, N - 3, \ldots, 2, 1, 0$

$$f_{N-1,N}(x_{N-1}, x_N) = v_N(x_{N-1}, x_N).$$

Exemplul 6.1. Pentru a asigura programul de fabricaţie, serviciul de aprovizionare al unei întreprinderi trebuie să achiziţioneze la fiecare 4 luni o anumită materie primă în cantităţi ce trebuie determinate. Preţul de achiziţie în unităţi monetare şi cererea în unităţi materie primă sînt date în tabelul următor:

Tabelul 6.1.

Perioada	i	1	2	3
Cererea	d_i	4	9	6
Preţul	p_i	19	10	12

Capacitatea de stocare a materiei prime este limitată la $S = 20$ unităţi.

Se cere să se determine cantităţile cu care să se aprovizioneze întreprinderea la începutul fiecărei perioade, astfel încît costul total de aprovizionare pe un an să fie minim în următoarele ipoteze:

a) stocul iniţial este $s_0 = 0$ iar stocul final $s_f = 4$;

b) stocul iniţial este $s_0 = 2$ stocul final nu este fixat.

Soluţie (*a*) Notăm cu x_i stocul la momentul i, a_i cantitatea cu care se aprovizionează întreprinderea la începutul perioadei i şi cu y_i stocul după aprovizionare în perioada i. Atunci trebuie să fie respectate următoarele condiţii pentru variabilele x_i care sînt variabile de stare şi de decizie.

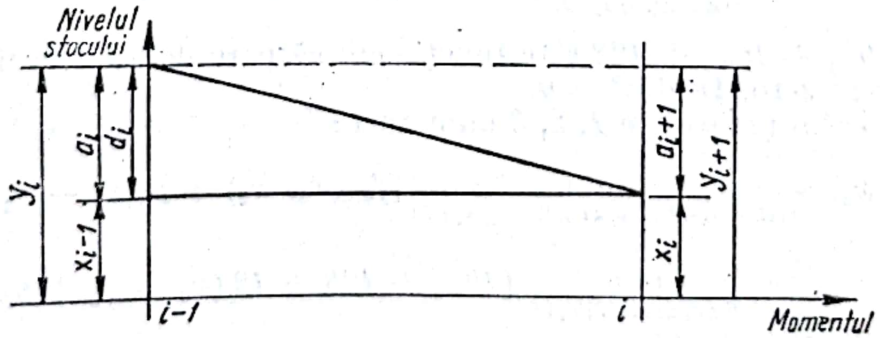


Fig. 6.1.

Avem, $0 \leqslant y_i \leqslant S$ şi $y_i = x_i + d_i$ deci $x_i \leqslant S - d_i$.

Pe de altă parte $a_i = y_i - x_{i-1} = x_i - x_{i-1} + d_i$ şi cum $a_i \geqslant 0$ avem

$$x_i - x_{i-1} + d_i \geqslant 0 \quad \text{adică}$$

$$x_i \leqslant x_{i+1} + d_{i+1}$$

Scanned with OKEN Scanner

Deci :

$$x_i \leqslant \min\,(S - d_i,\ x_{i+1} + d_{i+1}).$$

Cum $x_i \geqslant 0$ și $x_i \geqslant x_{i-1} - d_i$ avem de asemenea

$$x_i \geqslant \max\ (0,\ x_{i-1} - d_i)$$

În concluzie :

$$\max\,(0,\ x_{i-1} - d_i) \leqslant x_i \leqslant \min\,(S - d_i,\ x_{i+1} + d_{i+1})$$

$$i = 1,\ 2,\ \ldots,\ N - 1.$$

Trebuie deci determinate cantitățile

$$a_i = x_i - x_{i-1} + d_i \qquad i = 1,\ 2,\ 3,$$

astfel încît să fie minimă funcția cheltuielilor totale.

$$(\min)\ f = \sum_{i=1}^{3} p_i(x_i - x_{i-1} + d_i).$$

Să efectuăm optimizarea pe perioade în sensul trecut-viitor*.

$$f_{0,1}(x_0,\ x_1) = p_1(x_1 - x_0 + d_1) = 19(x_1 - 2 + 4) = 19x_1 + 38.$$

Luăm perioadele *1* și *2* împreună :

$$f_{0,2}(x_0,\ x_2) = \min_{\max(0,\ x_0 - d_1) \leqslant x_1 \leqslant \min(S - d_1,\ x_2 + d_2)} [f_{0,1}(x_0, x_1) + p_2(x_2 - x_1 + d_2)]$$

$$= \min_{0 \leqslant x_1 \leqslant \min(16,\ x_2 + 9)} [19x_1 + 38 + 10(x_2 - x_1 + 9)]$$

$$= \min_{0 \leqslant x_1 \leqslant (16;\ x_2 + 9)} [9x_1 + 10x_2 + 128] = 10x_2 + 128.$$

deoarece $9x_1 + 10x_2 + 128$ este funcție crescătoare de x_1 și deci minimul este atins în zero. Deci $x_1^* = 0$.

Considerăm perioadele *1*, *2*, *3* împreună :

$$f_{0,3}(x_0,\ x_3) = \min_{\max(0,\ x_1 - d_2) \leqslant x_2 \leqslant \min(S - d_2,\ x_3 + d_3)} [f_{0,2}(x_0,\ x_2) + p_3(x_3 - x_2 + d_3)]$$

$$= \min_{0 \leqslant x_2 \leqslant \min(11\,;\ x_3 + 6)} [10x_2 + 128 + 12\,(x_3 - x_2 + 6)]$$

$$= \min_{0 \leqslant x_2 \leqslant \min(11\,;\ x_3 + 6)} [-\,2x_2 + 12x_3 + 200]$$

---

* Adesea sensul „trecut-viitor" este numit „prospectiv", iar sensul „viitor-trecut" este numit „retrospectiv".

Scanned with OKEN Scanner

Cum x_3 reprezintă stocul final și $x_3 = 4$, obținem

$$f_{0:3}(x_0, x_3) = \min_{0 \leq x_2 \leq 10} [-2x_2 + 248] = 228$$

și $x_2^* = 10$.
Așadar politica optimă este dată de

$$x_0^* = 2,\ x_1^* = 0;\ x_2^* = 10;\ x_3^* = 4$$

sau în cantitățile ce trebuie achiziționate

$$a_1^* = 2;\ a_2^* = 19;\ a_3^* = 0$$

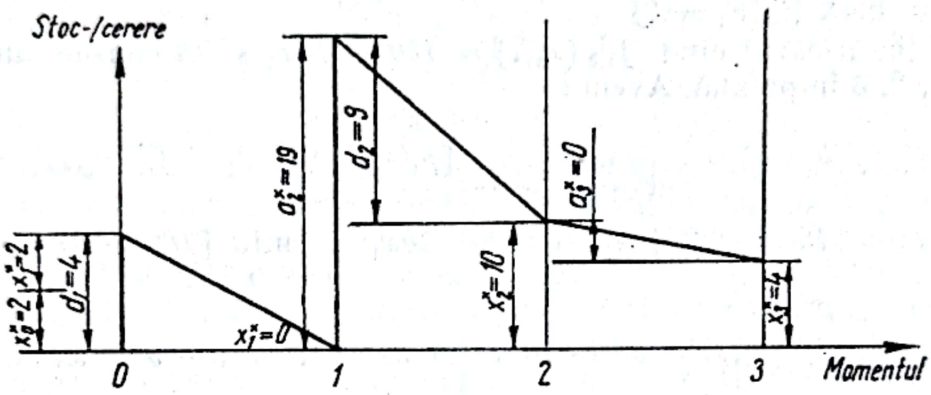


Fig. 6.2.

b) Să rezolvăm problema cînd stocul inițial este $s_0 = 2$ iar stocul final nu este fixat.
Vom face optimizarea în sensul viitor-trecut (retrospectiv)
Atunci

$$f_{2.3}(x_2, \cdot) = \min_{\max(0, x_2 - d_3) \leq x_3 \leq S - d_3} f_{2.3}(x_2, x_3)$$

$$= \min_{\max(0, x_2 - 6) \leq x_3 \leq x_3 \leq 14} [12(x_3 - x_2 + 6)]$$

Cum $12x_3 - 12x_2 + 72$ este funcție crescătoare de x_3 minimul este atins pentru $x_3 = \max(0, x_2 - 6)$.
Deci $f_{2:3}(x_2, \cdot) = 12 \max(0, x_2 - 6) - 12x_2 + 72$

$$= \max(0, 12x_2 - 72) - 12x_2 + 72$$

$$= 72 + \max(-12x_2, -72)$$

Luăm perioadele 2 și 3 împreună și obținem

$$f_{1.3}(x_1, \cdot) = \min_{\max(0, x_1 - d_2) \leq x_2 \leq S - d_2} [p_2(x_2 - x_1 + d_2) + f_{2.3}(x_2, \cdot)]$$

$$= \min_{\max(0, x_1 - 9) \leq x_2 \leq 11} [10(x_2 - x_1 + 9) + 72 + \max(-12x_2, -72)]$$

$$= \min_{\max(0, x_1 - 9) \leq x_2 \leq 11} [162 - 10x_1 + \max(-2x_2, 10x_2 - 72)]$$

Scanned with OKEN Scanner

Dacă $x_2 \leqslant 6$ atunci $-2x_2 \geqslant 10x_2 - 72$ şi deci

$$f^{(1)}_{1,3}(x_1,.) = \min_{\max(0,\, x_1-9) \leqslant x_2 \leqslant 6} [162 - 10x_1 - 2x_2] = 150 - 10x_1$$

cu $x_2^* = 6$
Dacă $x_2 \geqslant 6$ atunci $-2x_2 \leqslant 10x_2 - 72$
şi deci

$$f^{(2)}_{1,3}(x_1,.) = \min_{\max(6,\, x_1-9) \leqslant x_2 \leqslant 11} [162 - 10x_1 + 10x_2 - 72] =$$

$$= 90 - 10x_1 + 10 \max (6,\ x_1 - 9)$$

cu $x_2 = \max (6,\ x_1 - 9)$

Să luăm cazul cînd $f^{(1)}_{1,3}(x_1,.) = 150 - 10x_1$ şi să considerăm perioadele *1, 2, 3* împreună. Avem :

$$f^{(1)}_{0,3}(x_0,.) = \min_{\max(0,x_0-d_1) \leqslant x_1 \leqslant 8-d_1} [p_1(x_1 - x_0 + d_1) + f^{(1)}_{1,3}(x_1,.)]$$

$$= \min_{0 \leqslant x_1 \leqslant 16} [19(x_1 - 2 + 4) + 150 + 10x_1] = \min_{0 \leqslant x_1 \leqslant 16} [186 + 9x_1] = 186$$

cu $x_1^{*(1)} = 0$
Dacă luăm $f^{(2)}_{1,3}(x_1,.) = 90 - 10x_1 + \max (60,\ 10x_1 - 90)$ avem :

$$f^{(2)}_{0,3}(x_0,.) = \min_{0 \leqslant x_1 \leqslant 16} [19(x_1 + 2) + 90 - 10x_1 + \max (60,\ 10x_1 - 90)] =$$

$$= \min_{0 \leqslant x_1 \leqslant 16} [126 + 9x_1 + \max (60,\ 10x_1 - 90)]$$

Dacă $x_1 \leqslant 15$ atunci $60 \geqslant 10x_1 - 90$ şi deci

$$f^{(2,1)}_{0,3}(x_0,.) = \min_{0 \leqslant x_1 \leqslant 15} [126 + 9x_1 + 60] = 186$$

cu $x_1^{*(2)} = 0$
Dacă $x_1 \geqslant 15$ atunci $60 \leqslant 10x_1 - 90$ şi deci

$$f^{(2,2)}_{0,3}(x_0,.) = \min_{15 \leqslant x_1 \leqslant 16} [126 + 9x_1 + 10x_1 - 90] = 321$$

situaţie care nu convine întrucît :

$$f^{(2,2)}_{0,3}(x_0,.) > f^{(2,1)}_{0,3}(x_0,.) = f^{(1)}_{0,3}(x_0,.)$$

Urmează că $x_1^* = 0$ şi deci politica optimală

$$x_0^* = 2;\ \ x_1^* = 0;\ \ x_2^* = 6;\ \ x_3^* = 0$$

sau în cantităţile ce trebuie achiziţionate :

$$a_1^* = 2;\ \ a_2^* = 15;\ \ a_3^* = 0$$

Scanned with OKEN Scanner

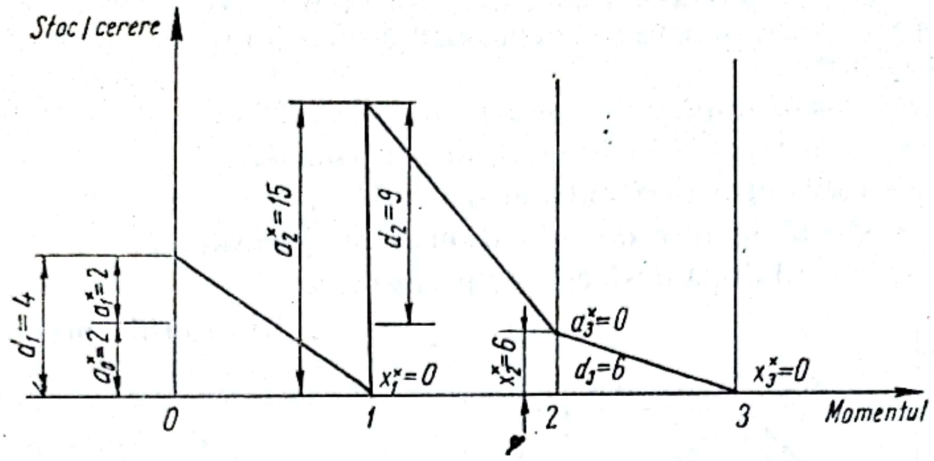


Fig. 6.3.

Se constată că cheltuielile totale de aprovizionare sînt mai mici decît în cazul cînd se impunea la sfîrşit existenţa stocului final $S_f = 4$ unităţi, ceea ce este natural deoarece aceasta este o restricţie suplimentară ceea ce face ca minimul să crească sau să stea pe loc.

Exemplul 6.2. O unitate de aprovizionare-desfacere trebuie să decidă periodic atît asupra cantităţilor cu care se aprovizionează cît şi asupra cantităţilor pe care le desface timp de un an, în fiecare trimestru, în condiţiile de mai jos :

— preţul de achiziţie şi preţul de desfacere variază trimestrial după tabelul de mai jos;

Tabelul 6.2.

i	$p_i^{(a)}$	$p_i^{(d)}$
1	5	9
2	9	10
3	17	17
4	10	13

— pentru lansarea unei comenzi se fac cheltuieli speciale care sînt aceleaşi oricare ar fi cantitatea comandată şi anume egale cu *16* unităţi monetare;

— capacitatea de stocare este limitată de $S = 15$ unităţi marfă;

— desfacerea în fiecare perioadă se presupune că se efectuează după ce s-a făcut aprovizionarea;

— conducerea unităţii decide asupra cantităţii cu care se aprovizionează şi pe care o desface în ipoteza că stocul iniţial $S_0 = 5$ şi stocul final $S_f = 0$.

Se cere să se stabilească politica de aprovizionare şi desfacere care asigură un beneficiu maxim unităţii.

Soluţie. Utilizăm un model dinamic discret în viitor sigur cu orizont limitat în care optimizarea se face de la viitor la trecut.

Scanned with OKEN Scanner

În fiecare perioadă avem două activităţi : o activitate de desfacere şi o activitate de achiziţionare aşa cum rezultă din schiţa de mai jos, cu notaţiile următoare :

y_i — stocul după achiziţionare în perioada i ;
a_i — cantitatea achiziţionată în perioada i ;
d_i — cantitatea desfăcută în perioada i ;
x_{i-1} — stocul înainte de achiziţionare în perioada i ;
x_i — stocul după desfacere în perioada i.

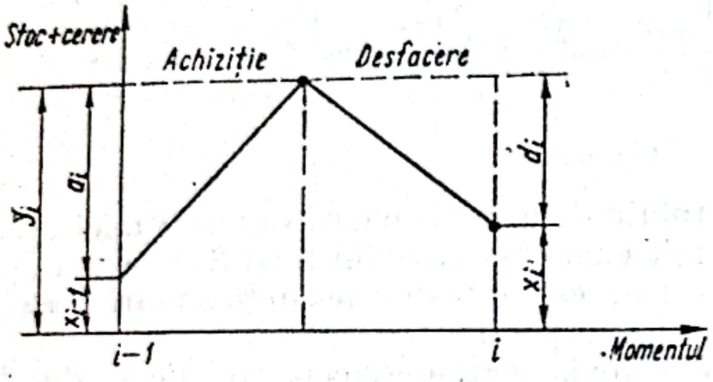


Fig. 6.4.

Restricţiile problemei :

$$a_i = y_i - x_{i-1}$$

$$d_i = y_i - x_i$$

Deci variabilele de stare şi decizie în fiecare perioadă i sînt x_{i-1}, x_i şi y_i pentru care avem restricţiile

$$x_{i-1} \leqslant y_i \leqslant S.$$

$$x_i \leqslant y_i \leqslant S.$$

deci

$$\max (x_{i-1}, x_i) \leqslant y_i \leqslant S.$$

De asemenea

$$0 \leqslant x_i \leqslant S$$

$$x_i \leqslant y_i$$

$$x_i \leqslant y_{i+1}$$

deci

$$0 \leqslant x_i \leqslant \min (y_i, y_{i+1})$$

Cheltuielile fixe de aprovizionare sînt funcţie de cantitatea achiziţionată :

$$d(a) = \begin{cases} 16, & a > 0 \\ 0, & a = 0. \end{cases}$$

Să trecem acum la determinarea politicii optimale.

În fiecare perioadă vom considera mai întîi latura desfacerii apoi latura aprovizionării (deoarece facem optimizarea în sensul viitor trecut).

Pentru perioada 4 notăm $g_4(y_4)$ încasările ca urmare a desfacerii. Atunci

$$g_4(y_4) = p_4^{(d)} (y_4 - x_4) = 13y_4$$

căci $x_4^* = 0$ dat prin condiţiile problemei :

Urmează că beneficiul obținut în perioada 4 este

$$f_{3,4}(x_3) = \max_{\max(x_3,\, x_4) \leqslant y_4 \leqslant S} [p_4^{(d)}(y_4 - x_4) - p_4^{(a)}(y_4 - x_3) - d(y_4 - x_3)]$$

$$= \max_{x_3 \leqslant y_4 \leqslant 15} [13y_4 - 10(y_4 - x_3) - d(y_4 - x_3)]$$

(intrucît $\max(x_3, x_4) = x_3$, x_4 fiind egal cu zero și $x_3 \geqslant 0$).

Deci $f_{3,4}(x_3) = \max_{x_3 \leqslant y_4 \leqslant 15} [3y_4 + 10x_3 - d(y_4 - x_3)] = \max_{y_4 = x_3,\, y_4 = 15} \{13x_3, 29 + 10x_3\}$

rămînînd nedeterminată valoarea lui y_4.

Activitatea de desfacere în perioada 3 ne conduce la încasările

$$g_3(y_3) = \max_{0 \leqslant x_3 \leqslant \min(y_3, y_4)} [\max \{13x_3, 29 + 10x_3\} + 17(y_3 - x_3)] =$$

$$= \max_{0 \leqslant x_3 \leqslant \min(y_3, y_4)} [\max \{-4x_3, 29 - 7x_3\} + 17y_3] = \max_{0 \leqslant x_3 \leqslant y_3} [29 - 7x_3 + 17y_3]$$

întrucît $\max \{-4x_3, 29 - 7x_3\} = 29 - 7x_3$ care corespunde lui $y_4^* = 15$.

Deci $g_3(y_3) = 29 + 17y_3$ corespunzător lui $x_3^* = 0$.

Să determinăm beneficiul maxim pentru perioadele 3 și 4 împreună:

$$f_{2,4}(x_2) = \max_{\max(x_2,\, x_3) \leqslant y_3 \leqslant 15} [29 + 17y_3 - 17(y_3 - x_2) - d(y_3 - x_2)]$$

$$= \max_{x_2 \leqslant y_3 \leqslant 15} [29 + 17x_2 - d(y_3 - x_2)] = 29 + 17x_2$$

corespunzător lui $y_3 = x_2$.

Încasările în perioada 2 sînt:

$$g_2(y_2) = \max_{0 \leqslant x_2 \leqslant \min(y_2,\, y_3)} [29 + 17x_2 + 10(y_2 - x_2)] = \max_{0 \leqslant x_2 \leqslant \min(y_2,\, x_2)} [29 + 7x_2 + 10y_2] =$$

$$= 29 + 17y_2$$

cu $x_2 = y_2$

Beneficiul maxim în perioadele 2, 3 și 4 va fi

$$f_{1,4}(x_1) = \max_{\max(x_1,\, x_2) \leqslant y_2 \leqslant 15} [29 + 17y_2 - 9(y_2 - x_1) - d(y_2 - x_1)]$$

$$= \max_{\max(x_1,\, x_2) \leqslant y_2 \leqslant 15} [29 + 9x_1 + [8y_2 - d(y_2 - x_1)] =$$

$$= \max \{29 + 17x_1, 133 + 9x_1\} =$$

$$= \begin{cases} 133 + 9x_1, \text{ dacă } 0 \leqslant x_1 \leqslant 13 \text{ cu } y_2 = 15 \\ 29 + 17x_1, \text{ dacă } 13 \leqslant x_1 \leqslant 15 \text{ cu } y_2 = x_1 \end{cases}$$

Încasările în perioada 1 sînt:

Dacă $0 \leqslant x_1 \leqslant 13$

$$g_1(y_1) = \max_{0 \leqslant x_1^* \leqslant \min(13,\, \min(y_1, 15))} [133 + 9x_1 + 9(y_1 - x_1)] = \max_{0 \leqslant x_1 \leqslant \min(13,\, y_1)} [133 + 9y_1] = 133 + 9y_1$$

oricare ar fi $x_1 \in [0, \min(13, y_1)]$

Scanned with OKEN Scanner

iar beneficiul maxim pe toate cele patru perioade va fi :

$$f_{0;4}(x_0) = \max_{\max(x_0,\, x_1) \leqslant y_1 \leqslant 15} [133 + 9y_1 - 5(y_1 - x_0) - d(y_1 - x_0)]$$

$$= \max_{\max(5,\, x_1) \leqslant y_1 \leqslant 15} [133 + 9y_1 - 5y_1 + 25 - d(y_1 - 5)] =$$

$$= \max_{\max(5,\, x_1) \leqslant y_1 \leqslant 15} [158 + 4y_1 - d(y_1 - 5)] = 202.$$

Dacă $13 \leqslant x_1 \leqslant 15$, atunci

$$g_1(y_1) = \max_{13 \leqslant x_1 \leqslant \min(y_1,\, y_2)} [29 + 17x_1 + 9(y_1 - x_1)] = \max_{13 \leqslant x_1 \leqslant \min(y_1,\, x_1)} [29 + 17x_1 + 9y_1 - 9x_1] =$$

$$= \max_{13 \leqslant x_1 \leqslant y_1} [29 + 8x_1 + 9y_1] = 29 + 17y_1$$

cu $x_1 = y_1$.

Urmează în acest caz că :

$$f_{0;4}(x_0) = \max_{\max(x_0,\, x_1) \leqslant y_1 \leqslant 15} [29 + 17y_1 - 5(y_1 - x_0) - d(y_1 - x_0)] =$$

$$= \max_{13 \leqslant x_1 \leqslant y_1 \leqslant 15} [29 + 17y_1 - 5y_1 + 25 - d(y_1 - 5)] =$$

$$= 218, \text{ cu } y_1^* = 15$$

Urmează deci că

$$x_1^* = 15\ ;\ y_2^* = 15;\ x_2^* = 15;\ y_3^* = 15$$

$$x_3^* = 0;\ y_4^* = 15;\ x_4^* = 0$$

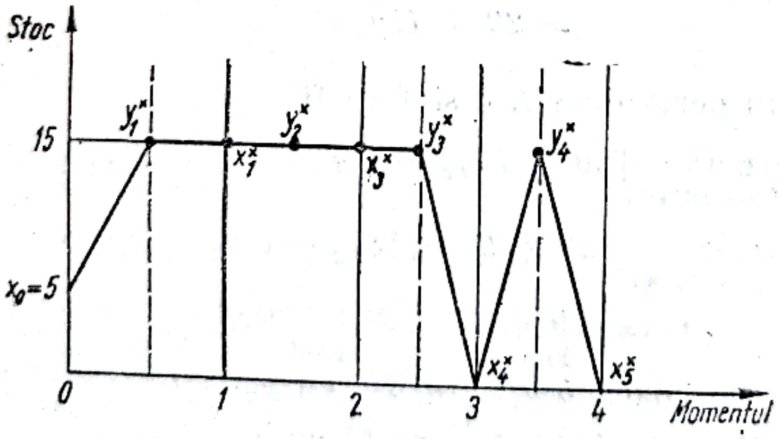


Fig. 6.5.

$$a_1^* = y_1^* - x_0^* = 15 - 5 = 10;\ d_1^* = y_1^* - x_1^* = 15 - 15 = 0$$

$$a_2^* = y_2^* - x_1^* = 15 - 15 = 0;\ d_2^* = y_2^* - x_2^* = 15 - 15 = 0$$

$$a_3^* = y_3^* - x_2^* = 15 - 15 = 0;\ d_3^* = y_3^* - x_3^* = 15 - 0 = 15$$

$$a_4^* = y_4^* - x_3^* = 15 - 0 = 15;\ d_4^* = y_4^* - x_4^* = 15 - 0 = 15.$$

Scanned with OKEN Scanner

6.3. Programe dinamice staționare

Spunem că un program dinamic în viitor sigur (determinist) este „staționar" dacă pentru toate valorile n ale perioadelor pe care se desfășoară programul — cu excepția eventuală a valorilor extreme — sînt satisfăcute relațiile :

$$\Gamma_n x_{n-1} = \Gamma x_{n-1}$$

$$v_n(x_{n-1},\ x_n) = v(x_{n-1},\ x_n).$$

Din definiția dată rezultă că pentru un program dinamic staționar aplicațiile Γ_n ce definesc restricțiile programului precum și valorile atribuite fiecărei perioade $v_n(x_{n-1},\ x_n)$ sînt independente de perioada n considerată.

Bazați pe acestea putem scrie :

$$\mathbf{X}_n(x_0,\ x_N) = \Gamma^{(n)}x_0 \cap \Gamma^{-(N-n)}x_N$$

unde

$$\Gamma^{(n)}x_0 = \Gamma(\Gamma^{(n-1)}x_0) \quad \text{și} \quad \Gamma^{-(N-n)}x_N = \Gamma^{-1}(\Gamma^{-(N-n+1)}x_N).$$

Aceasta conduce la faptul că oricare ar fi k, cu condiția ca perioadele ce intervin în considerație să facă parte din perioadele pe care este definit programul dinamic, vom avea :

$$\mathbf{X}_{n+k}(x_k,\ x_{n'+k}) = \Gamma^{(n)}x_k \cap \Gamma^{-(n'-n)}x_{n'+k}$$

De aici rezultă că expresia :

$$\mathbf{X}_{n+k}(x_k = x,\ x_{n'+k} = x') = \mathbf{X}_n(x_0 = x,\ x_{n'} = x')$$

este independentă de k.

Dacă pentru simplificarea scrierii vom introduce notația

$$\mathbf{X}_{n+k}(x_k = x,\ x_{n'+k} = x') = \mathbf{X}_{n,\,n'-n}(x,\ x')$$

și

$$f_k(x,\ x') = f_{n,\,n+k}(x,\ x')$$

atunci sistemele de relații de recurență cu ajutorul cărora punem în evidență politicile optimale devin

$$f_n(x,\ x'') = \operatorname*{opt}_{x' \in \mathbf{X}_{n-1,1}(x,\ x'')} [f_{n-1}(x,\ x') + v(x',\ x'')]$$

$$n = 2,\ 3, \ldots, N$$

$$f_1(x,\ x') = v(x,\ x')$$

și respectiv

$$f_n(x,\ x'') = \operatorname*{opt}_{x' \in \mathbf{X}_{1,\,n-1}(x,\ x'')} [v(x,\ x') + f_{n-1}(x',\ x'')]$$

$$n = 2, 3, \ldots, N$$

$$f_1(x_1,\ x') = v(x_1 x').$$

Scanned with OKEN Scanner

6.4. Model dinamic de programare optimală a investițiilor

Ca aplicație generală a utilizării programelor dinamice în rezolvarea unor probleme economice concrete vom considera o problemă de repartizare optimă a investițiilor.

Să luăm următoarea problemă de repartizare a investițiilor :

În k regiuni economice se intenționează să se facă o investiție în valoare totală de I unități monetare (unitatea monetară este exprimată în milioane, zeci de milioane etc.).

O unitate monetară investită aduce un beneficiu care diferă în funcție de suma investită și de asemenea de regiunea în care s-a făcut investiția. Se cere să se determine o repartizare a investițiilor pe regiuni astfel încît pe ansamblul regiunilor să se obțină un beneficiu maxim.

Am putea aborda problema luînd în considerație toate combinațiile posibile și să alegem apoi acea politică (sau acele politici dacă soluția nu este unică) care asigură maximul funcției de eficiență. Această cale nu este practică deoarece conduce la un număr extraordinar de mare de politici dacă I și k sînt mari. Așa de exemplu dacă $\mathrm{I} = 10$ și $k = 4$ se obțin *286* de politici din care trebuie aleasă politica optimă, ceea ce constituie o treabă dificilă chiar utilizînd calculatorul.

Dar, dacă vom urmări o tratare secvențială a acestei probleme vom obține incomparabil mai ușor soluția optimă. Să considerăm beneficiile corespunzătoare celor k regiuni date în tabelul de mai jos.

Tabelul 6.3.

Investiții în unități	Beneficiul regiunii *1*	Beneficiul regiunii *2*	...	Beneficiul regiunii k
0	$v_1(0)$	$v_2(0)$	...	$v_k(0)$
1	$v_1(1)$	$v_2(1)$	...	$v_k(1)$
2	$v_1(2)$	$v_2(2)$	...	$v_k(2)$
⋮	⋮	⋮		⋮
I	$v_1(\mathrm{I})$	$v_2(\mathrm{I})$	...	$v_k(\mathrm{I})$

Deci, dacă notăm cu $x_1, x_2, \ldots, x_k$ investițiile exprimate în unitatea aleasă, în regiunea *1*, regiunea *2*, ș.a.m.d. în regiunea k, beneficiile pe regiuni vor fi $v_1(x_1), v_2(x_2), \ldots, v_k(x_k)$ respectiv. Notînd cu $\mathrm{F}(x_1, x_2, \ldots, x_k)$ beneficiul total, rezultă că

$$\mathrm{F}(x_1, x_2, \ldots, x_k) = \sum_{i=1}^{k} v_i(x_i)$$

cu restricțiile

$$\sum_{i=1}^{k} x_i = \mathrm{I},\ x_i = 0, 1, 2, \ldots, \mathrm{I},\ i = 1, 2, \ldots, k.$$

Scanned with OKEN Scanner

Urmează că trebuie rezolvată problema

$$\max \mathbf{F}(x_1, x_2, \ldots, x_k)$$
$$\sum_{i=1}^{k} x_i = \mathrm{I}$$
$$x_i = 0,1,2,\ldots,\mathrm{I}$$
$$i = 1, 2, \ldots, k$$

Vrem să tratăm secvenţial această problemă şi pentru aceasta procedăm după cum urmează:

Punem:

$$x_1 + x_2 = u_1$$
$$u_1 + x_3 = u_2$$
$$u_2 + x_4 = u_3$$
$$\ldots\ldots\ldots\ldots$$
$$u_{n-2} + x_k = u_{k-1} \leqslant \mathrm{I}.$$

Urmează că $u_1 \leqslant u_2 \leqslant \ldots \leqslant u_{k-2} \leqslant u_{k-1} \leqslant \mathrm{I}$, iar u_{k-1} ia succesiv valorile 0, 1, 2, ..., I.

Atunci $\mathbf{F}(x_1, x_2, \ldots, x_k)$ se mai poate scrie

$$\mathbf{F}(x_1, x_2, \ldots, x_k) = \mathbf{F}(x_1, u_1, u_2, \ldots, u_{k-2}, u_{k-1}) =$$
$$= v_1(x_1) + v_2(u_1 - x_1) + v_3(u_2 - u_1) + \ldots + v_k(u_{k-1} - u_{k-2})$$

De aici rezultă că putem utiliza un proces iterativ calculînd succesiv

$$f_{0,1}(x_1) = v_1(x_1)$$
$$f_{0,2}(u_1) = \max_{x_1 = 0, 1, \ldots, u_1} [v_1(x_1) + v_2(u_1 - x_1)]$$
$$f_{0,3}(u_2) = \max_{u_1 = 0, 1, 2, \ldots, u_2} [f_{0,2}(u_1) + v_3(u_2 - u_1)]$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$f_{0;\,k-1}(u_{k-2}) = \max_{u_{k-3} = 0, 1, \ldots, u_{k-2}} [f_{0,\,k-2}(u_{k-3}) + v_{k-1}(u_{k-2} - u_{k-3})]$$
$$f_{0,k}(\mathrm{I}) = \max_{u_{k-1} = 0, 1, 2, \ldots, \mathrm{I}} [f_{0,\,k-1}(u_{k-2}) + v_k(\mathrm{I} - u_{k-2})]$$

Urmează că

$$\max \mathbf{F}(x_1, x_2, \ldots, x_k) = f_{0,k}(\mathrm{I}).$$
$$\sum_{i=1}^{k} x_i = \mathrm{I};\ x_i = 0, 1, \ldots, \mathrm{I};\ i = 1, 2, \ldots, k$$

Exemplul 6.3. Se cere să se stabilească o repartizare optimă a investiţiei de $\mathrm{I} = 50$ milioane lei în trei sectoare economice considerînd că

Scanned with OKEN Scanner

volumul minim al investițiilor pentru fiecare obiectiv este de 10 milioane lei și știind că beneficiile unitare sînt date în tabelul de mai jos.

Tabelul 6.4.

Sectorul / Suma investită x_i, u, m	S_1	S_2	S_3
0	0	0	0
1	0,3	0,25	0,25
2	0,5	0,45	0,30
3	0,70	0,60	0,45
4	0,80	0,65	0,50
5	0,90	0,80	0,65

Soluția. S-a ales ca unitate monetară 10 milioane lei.
În prima etapă se aplică relația :

$$f_{0,2}(u_1) = \max_{x_1 = 0,\ 1,\ \ldots,\ u_1} [v_1(x_1) + v_2(u_1 - x_1)]$$

$$u_1 = 0,\ 1,\ 2,\ 3,\ 4,\ 5.$$

Rezultatele le trecem sintetic în tabelul 6.5.

Tabelul 6.5.

Investiții	$v_1(x_1)$	$v_2(x_2)$	$f_{0,2}(u_1)$	Politica optimă pentru S_1 și S_2
0	0	0	0	(0,0)
1	0,3	0,25	0,3	(1,0)
2	0,5	0,45	0,55	(1,1)
3	0,7	0,6	0,75	(1,2), (2,1)
4	0,8	0,65	0,95	(2,2); (3,1)
5	0,9	0,8	1,15	(3,2)

În a doua etapă se consideră și cea de a treia regiune economică, aplicîndu-se relația :

$$f_{0,3}(u_2) = \max_{u_1 = 0,\ 1,\ \ldots,\ u_2} [f_{0,2}(u_1) + v_3(u_2 - u_1)]$$

$$u_2 = 0,\ 1,\ 2, \ldots,\ 5.$$

Obținem astfel :

Tabelul 6.6.

Investiția	$f_{0,2}(u_1)$	$v_3(x_3)$	$f_{0,3}(u_2)$	Politica optimă pentru S_1, S_2, S_3
0	0	0	0	(0,0,0)
1	0,3	0,25	0,3	(1,0,0)
2	0,55	0,30	0,55	(1,0,1); (1,1,0)
3	0,75	0,45	0,80	(1,1,1)
4	0,95	0,50	1,00	(1,2,1); (2,1,1)
5	1,15	0,65	1,20	(2,2,1); (3,1,1)

Scanned with OKEN Scanner

Urmează că beneficiul maxim se obţine pentru următoarele două politici de investiţii

$x_1 = 20$ milioane în sectorul S_1
$x_2 = 20$ milioane în sectorul S_2
$x_3 = 10$ milioane în sectorul S_3

sau

$x_1 = 30$ milioane în sectorul S_1
$x_2 = 10$ milioane în sectorul S_2
$x_3 = 10$ milioane în sectorul S_3.

6.5. Interpretări ale programelor dinamice cu ajutorul teoriei grafelor. Grafe secvenţiale

După cum am văzut se numeşte graf cuplul $\mathbf{G} = (\mathbf{X}, \Gamma)$ unde $\mathbf{X}$ este o mulţime de puncte numite vîrfuri iar Γ o aplicaţie multivocă în general, a lui $\mathbf{X}$ în $\mathbf{X}$.

În contextul celor studiate de noi — adică pentru programe dinamice discrete în viitor sigur cu orizont limitat

$$\mathbf{X} = \bigcup_{n=o}^{N} \mathbf{X}_n(x_0,\ x_N)$$

iar Γ este definită prin

$$\Gamma\tilde{x}_{n-1} = \Gamma_n\tilde{x}_{n-1} \cap \mathbf{X}_n(x_0,\ x_N)$$

unde $\tilde{x}_{n-1}$ este o valoare particulară a variabilei de stare x_{n-1} astfel încît

$$\tilde{x}_{n-1} \in \mathbf{X}_{n-1}\ (x_0,\ x_N)$$

Vom presupune mai departe că variabila de stare x_n este unidimensională şi că mulţimile $\mathbf{X}_n$ ale valorilor posibile sînt cel mult numărabile.

Datorită condiţiilor impuse, rezultă că elementele mulţimii $\mathbf{X}$ pot fi reprezentate prin puncte (adică valorile $\tilde{x}_n$ ale variabilei de stare x_n).

Relaţia $\tilde{x}_n \in \Gamma_n\tilde{x}_{n-1}$ este reprezentată printr-un arc ce uneşte vîrful $\tilde{x}_{n-1}$, cu vîrful $\tilde{x}_n$ orientat în sensul de la $\tilde{x}_{n-1}$ la $\tilde{x}_n$.

Ţinînd seama de definiţia unei reţele de transport constatăm că un program dinamic corespunde unei reţele de transport fără circuite în care valoarea asociată arcului $(\tilde{x}_{n-1},\ \tilde{x}_n)$ este $v_n(\tilde{x}_{n-1},\ \tilde{x}_n)$.

Este o reţea de transport particulară deoarece vîrfurile grafului formează $N + 1$ mulţimi $(\mathbf{X}_n)_{o]\leqslant n\leqslant N}$ cu proprietatea că plecînd dintr-un vîrf $\tilde{x}_n \in \mathbf{X}_n$ se merge obligatoriu într-un vîrf $\tilde{x}_{n+1} \in \mathbf{X}_{n+1}$.

O astfel de reţea o vom numi „reţea de transport secvenţială".

Să considerăm acum un graf finit și fără circuite $G(X, \Gamma)$ și să definim submulțimile $X_0, X_1, \ldots, X_N$ astfel încît :

$$X_0 = \{x_r : \Gamma^{-1}x_r = \Phi\}$$

$$X_1 = \{x_r : \Gamma^{-1}x_r \subset X_0\}$$

$$X_2 = \{x_r : \Gamma^{-1}x_r \subset X_0 \cup X_1\}$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$X_N = \{x_r : \Gamma^{-1}x_r \subset \bigcup_{n=0}^{N-1} X_n\}$$

unde N este cel mai mic număr natural pentru care este satisfăcută relația $\Gamma X_N = \Phi$.

Submulțimile de vîrfuri $X_0, X_1, \ldots, X_n$ constituie o partiție a mulțimii X, adică :

$$X_n \cap X_{n'} = \Phi \text{ dacă } n \neq n' \text{ și } \bigcup_{n=0}^{N} X_n = X,\ X_n \neq \Phi$$

și sînt total strict ordonate prin relația de ordine

$$X_n \prec X_{n'} \text{ dacă și numai dacă } n < n'^*$$

Prin definiție aplicația $\mathbf{0} : X \to \{0, 1, 2, \ldots, N\}$ definită prin :

$$x_r \in X_n \Rightarrow 0(x_r) = n$$

o vom numi „funcție ordinală a grafului".

Se cunosc mai multe metode de aflare a funcției ordinale a unui graf. Vom prezenta o metodă (A. Kaufman) care constă în următoarele :

Se consideră matricea booleană $B = (b_{ij})_{1 \leqslant i, j \leqslant M}$ atașată grafului, unde $M = |X|$ (numărul elementelor mulțimii X) ale cărei elemente b_{ij} sînt :

$$b_{ij} = \begin{cases} 1 \text{ dacă } x_j \in \Gamma x_i \\ 0 \text{ dacă } x_j \notin \Gamma x_i \end{cases}$$

și se ia apoi matricea transpusă B'.

Dacă notăm cu $V_1, V_2, \ldots, V_M$ vectorii coloană ai matricii B', $B' = (V_1, V_2, \ldots, V_M)$ considerăm vectorul $V^{(0)} = \sum_{k=1}^{M} V_k$

Mulțimea vîrfurilor corespunzătoare componentelor zero din vectorul $V^{(0)}$ constituie mulțimea X_0 și avem totodată valoarea „0" a funcției ordinale.

Mai departe se calculează :

$$V^{(1)} = V^{(0)} - (V_{k_1} + V_{k_2} + \ldots + V_{k_{r_0}})$$

unde V_{k_i} sînt vectorii corespunzători vîrfurilor ce fac parte din mulțimea X_0. Obținem de aici valoarea „1" a funcției ordinale și vîrfurile ce constituie mulțimea X_1 care este alcătuită din mulțimea vîrfurilor corespunzătoare componentelor zero din vectorul $V^{(1)}$ (altele decît cele obținute în etapa

---

* Relația $X_n \prec X_{n'}$, o vom citi „X_n precede strict pe $X_{n'}$".

precedentă). Procedăm astfel din aproape în aproape pînă ce s-au epuizat toate vîrfurile grafului, cînd se încheie procesul.

Putem sintetiza într-un tabel pașii algoritmului pentru determinarea funcției ordinale a unui graf (tabelul 6.7).

Definiție. Un graf $G(X, \Gamma)$ care satisface proprietățile

1) graful G are o funcție ordinală $O(x_r)$ cu valorile $0, 1, 2, \ldots, N$.

2) $\Gamma X_n \subset X_{n+1}$ unde $X_n = \{x_r : O(x_r) = n\}$, $n = 0, 1, 2, \ldots, N-1$, îl vom numi graf secvențial.

Tabelul 6.7.

	x_1	x_2	...	x_M	$V^{(0)}$	$V^{(1)}$	...	$V^{(N)}$
x_1	b'_{11}	b'_{12}	...	b'_{1M}	$\sum_{j=7}^{M} b'_{1j}$	$\sum_{j=1}^{M} b'_{1j} - \sum_{1=1}^{r_0} b'_{1k_1}$	...	
x_2	b'_{21}	b'_{22}	...	b'_{2M}	$\sum_{j=1}^{M} b'_{2j}$	$\sum_{j=1}^{M} b'_{2j} - \sum_{l=1}^{r_0} b'_{2k_l}$	...	
x_M	b'_{M1}	b'_{M2}	...	b'_{MM}	$\sum_{j=1}^{M} b'_{Mj}$	$\sum_{j=1}^{M} b_{Mj} - \sum_{l=1}^{r_0} b'_{Mk_l}$	...	
Valoarea funcției ordinale					0	1		N
Vîrfurile					$x_{k_1}, x_{k_2}, \ldots, x_{k_{n_0}}$			
Mulțimile ordonate de vîrfuri					X_0	X_1		X_N

Rezultă că dacă $G = (X, \Gamma)$ este un graf secvențial atunci au loc incluziunile

$$\Gamma^{-1} X_{n+1} \subset X_n$$

$$\Gamma X_n \subset X_{n+1} \quad n = 0, 1, 2, \ldots, N-1$$

și deci

$$\Gamma X_n = X_{n+1}$$

De aici urmează că orice subgraf $G_n = (X_n \cup X_{n+1}; \Gamma_{X_n \cup X_{n+1}})$ $n = 0, 1, 2, \ldots, N-1$, al grafului dat, este un graf simplu.

Reciproc, dacă într-un graf $G = (X, \Gamma)$ există o partiție $X_0, X_1, \ldots, X_N$ a mulțimii vîrfurilor X astfel încît orice subgraf de forma

$$G_n = (X_n \cup X_{n+1}, \Gamma_{X_n \cup X_{n+1}}) \quad n = 0, 1, \ldots, N-1,$$

să fie simplu, atunci graful G este secvențial.

Scanned with OKEN Scanner

Algoritmul descris anterior pentru aflarea funcției ordinale poate fi utilizat cu succes pentru a proba dacă un graf dat este un graf simplu.

Considerațiile făcute ne dau posibilitatea să afirmăm că graful asociat unui program dinamic este un graf secvențial și invers. Adică — fiind dat un sistem a cărui stare este caracterizată cu ajutorul unei variabile și care evoluează ca urmare a deciziilor luate — pentru ca deciziile să poată fi alese cu ajutorul programării dinamice după modelul discret, în viitor sigur cu orizont limitat trebuie ca graful asociat evoluțiilor posibile ale sistemului să fie secvențial.

Pentru un astfel de graf, tuturor vîrfurilor submulțimii $\mathbf{X}_n$ le vor corespunde un „moment" egal cu valoarea n a funcției ordinale iar cuplului de două momente succesive $(n, n+1)$ îi va corespunde o „perioadă" (perioada $n+1$). Deci perioada $n+1$ va corespunde subgrafului.

$$\mathbf{G}_n = (\mathbf{X}_n \cup \mathbf{X}_{n+1}, \mathbf{\Gamma}_{\mathbf{X}_n \cup \mathbf{X}_{n+1}})$$

6.6. Programe dinamice discrete în viitor sigur și orizont nelimitat

În programele dinamice pe care le-am considerat intervenea esențial faptul că numărul perioadelor este limitat la N.

Adesea însă în practica economică aceste restricții ridică obiecții deoarece cel puțin atunci cînd intervin programe staționare, orizontul economic nu mai este limitat de un număr determinat de perioade.

Să considerăm un program dinamic și să presupunem pentru a simplifica lucrurile că starea inițială x_0 este impusă și unică. Pentru ca acest program dinamic să fie considerat pe un număr nelimitat de perioade trebuie să fie satisfăcute condițiile

1) $\mathbf{\Gamma}_n x_{n-1}$ să fie definită pentru orice $n \geqslant 0$ și orice $x_{n-1} \in \mathbf{X}_{n-1}(x_0, .)$.

2) valoarea $v_n(x_{n-1}, x_n)$ să fie definită pentru orice $n > 0$, orice $x_{n-1} \in \mathbf{X}_{n-1}(x_0, .)$ și orice $x_n \in \mathbf{\Gamma}_n x_{n-1}$.

În aceste condiții o politică este o succesiune determinată a variabilelor de stare,

$$\pi = (x_0, x_1, \ldots, x_n, \ldots)$$

Va trebui să vedem ce înseamnă optimal, deoarece acum nu mai putem distinge politicile după valoarea lor finală ca în cazul unui număr N (dat) de perioade, căci în cazul unui număr nelimitat de perioade valoarea finală poate fi infinită. Ar rezulta că o politică optimală este politica de valoare infinită ceea ce nu prezintă interes practic, căci toate politicile cu valoare infinită ar fi considerate echivalente deși ele pot să difere esențial pe orice parte finită. Pentru a înlătura acest inconvenient se va căuta o politică

$$\pi^* = \{x_0, x_1^*, \ldots, x_n^*, \ldots\}$$

astfel încît subpolitica

$$\pi_{o,n}^* = \{x_0, x_1^*, \ldots, x_n^*\}$$

Scanned with OKEN Scanner

să fie pentru orice n sau cel puţin pentru orice $n > N$, optimală de la x_0 la n.

În cele ce urmează vom considera unele cazuri particulare tipice de programe dinamice staţionare pentru care definim o funcţie de valoare mărginită.

Să luăm mai întîi o clasă de programe dinamice staţionare în care variabila de decizie este unidimensională iar restricţiile sînt de forma:

$$\Gamma x_{n-1} = \{x_n : b\, x_{n-1} \leqslant x_n \leqslant a x_{n-1}\}$$

unde $0 \leqslant b < a < 1$.

Dacă punem $f_n(x,.) = f_n(x)$, atunci relaţia

$$f_n(x,.) = \max_{bx \leqslant x' \leqslant ax} \{v(x, x') + f_{n-1}(x',.)\}$$

va căpăta forma

$$f_n(x) = \max_{bx \leqslant x' \leqslant ax} \{v(x, x') + f_{n-1}(x')\}$$

cu $f_0(x) = 0$

Trecînd formal la limită pentru $n \to \infty$ se obţine ecuaţia funcţională:

$$f(x) = \max_{bx \leqslant x' \leqslant ax} \{v(x, x') + f(x')\}$$

Se pune problema existenţei soluţiei acestei ecuaţii şi de asemenea legătura ei cu programul dinamic considerat.

Răspunsul la problema considerată este dat de teorema pe care o enunţăm fără a da şi demonstraţia ei.

Teoremă. Fie $a, b \in R$ astfel încît $a \leqslant b < a < 1$

Dacă

1) $v(x, x')$ este continuă în raport cu ambele variabile

2) funcţia $u_0(x) = \max\limits_{0 \leqslant z \leqslant x} |\max\limits_{bz \leqslant z' \leqslant az} v(z, z')|$

verifică relaţia

$$\sum_{n=0}^{\infty} u_0 (a^n x) < \infty$$

pentru orice $x \geqslant 0$, atunci ecuaţia funcţională

$$f(x) = \max_{bx \leqslant x' \leqslant ax} [v(x, x') + f(x')]$$

are o soluţie unică $f(x)$, continuă şi care este nulă în $x = 0$.

Valoarea optimală a programului dinamic pe n perioade definit pe restricţiile $bx \leqslant x' \leqslant ax$ şi cu valoarea $v(x, x')$, este dată de

$$f_n(x) = \max_{bx \leqslant x' \leqslant ax} \{v(x, x') + f_{n-1}(x')\}$$

iar şirul $(f_n(x))_{n \in N^*}$ converge uniform către $f(x)$.

Scanned with OKEN Scanner

Observație: În teorema de existență și unicitate se ia $f_0(x) = 0$. Ea rămîne însă adevărată dacă se înlocuiește $f_0(x)$ cu această proprietate, cu o funcție arbitrară, continuă pentru $x > 0$ și nulă pentru $x = 0$.

Acest lucru permite să se scurteze procesul aproximațiilor succesive cu ajutorul cărora se obține funcția $f(x)$, dacă se cunoaște o funcție $f_0(x)$ apropiată de $f(x)$.

Prima decizie a unei politici optimale este dată de funcția $x'^*(x)$ care realizează maximul în relația:

$$f(x) = \max_{bx \leqslant x' \leqslant ax} [v(x, x') + f(x')]$$

și această funcție definește complet mulțimea politicilor optimale.

Definiție. Se numește „politică permanentă" acea politică pentru care decizia luată în cursul unei perioade la începutul căreia avem starea x, este $y(x)$, unde $y(x)$ este funcție univocă, independentă de perioada considerată și pe care o vom numi „funcție de decizie".

Procesul aproximațiilor succesive pentru aflarea funcției $f(x)$ se scurtează dacă se cunoaște o funcție $f_0(x)$ apropiată de $f(x)$, însă acest lucru este dificil de obținut.

Se poate determina totuși în general pe baza experienței sau intuiției o politică permanentă $y_0(x)$ apropiată de politica optimală $y(x)$. Funcția $f_0(x)$ corespunzătoare politicii $y_0(x)$ este dată de ecuația funcțională

$$f_0(x) = v\ [x, y_0(x)] + f_0(y_0(x))$$

Cu funcția $f_0(x)$ determinată astfel se trece la determinarea unei noi politici $y_1(x)$ pe baza relației:

$$f_1(x) = \max_{bx \leqslant y \leqslant ax} [v(x, y) + f_0(y)]$$

De aici rezultă că $f_1(x) \geqslant f_0(x)$ oricare ar fi x.

Dacă $f_1(x) = f_0(a)$ rezultă că $y_0(x)$ ales inițial este optimal și problema este rezolvată, iar dacă nu, se calculează mai departe $f_2(x)$, $f_3(x)$,... obținîndu-se în felul acesta un șir monoton crescător $\{f_n(x)\}$ de funcții care ne dau posibilitatea să obținem o politică optimală.

6.7. Criteriul valorii actuale

Programele dinamice în care convergența procesului de aproximare este asigurată prin contracția domeniului deciziilor posibile reprezintă un caz particular de programare dinamică în viitor sigur cu orizont nelimitat.

Cum adesea valoarea $f_n(x)$ a unei politici optimale pe n perioade devine infinită cînd $n \to \infty$ interesează aflarea unor procedee care să asigure convergența procesului.

Un astfel de procedeu îl constituie criteriul valorii actualizate, justificat în marea majoritate a cazurilor, de considerații economico-financiare.

Scanned with OKEN Scanner

După cum se știe suma s disponibilă peste n ani poate fi considerată ca echivalentă sumei disponibilă imediat

$$s' = \frac{s}{(1+1)^n}$$

dacă se presupune dobînda constantă pentru toți cei n ani și egală cu i procente.

Cu alte cuvinte împrumutînd suma s' putem dispune de ea imediat, dar vom avea de rambursat după n ani suma $s = (1 + i)^n s'$.

Valoarea s' se numește „valoare actuală" sau „valoare actualizată" a sumei s.

Introducînd parametrul $r = \frac{1}{1+i} < 1$, numit *coeficient de actualizare*, suma s' se mai poate scrie $s' = r^n s$.

Acum dacă într-un program dinamic „valoarea" atașată fiecărei perioade este o sumă de bani sau un echivalent, atunci este logic să comparăm două politici ale programului dinamic după valoarea lor actualizată.

Vom presupune că valoarea $v_n(x_{n-1}, x_n)$ a unei decizii luate în perioada n se subînțelege la momentul n, adică la sfîrșitul perioadei n. În felul acesta dacă convenim să luăm drept referință momentul 0, valoarea actuală a acestei decizii va fi $r^n v_n (x_{n-1}, x_n)$.

Să considerăm acum un program dinamic staționar în viitor sigur, cu orizont limitat și să introducem valoarea actualizată. Se vede imediat că prin intervenția valorilor actualizate noul model obținut nu va mai fi un model staționar ci semistaționar.

Luînd un program dinamic staționar și notînd cu $f(x, r)$ valoarea actualizată pe n perioade, vom putea scrie :

$$f_n(x, r) = r \cdot \max_{x' \in \Gamma x} [v(x, x') + f_{n-1}(x', r)]$$

Într-adevăr în primul membru $f_n(x, r)$ reprezintă valoarea optimală actualizată la începutul primei perioade din cele considerate. În membrul al doilea $\max_{x' \in \Gamma x} [v(x, x') + f_{n-1}(x', r)]$ reprezintă același lucru la sfîrșitul primei perioade, adică la începutul celei de a doua perioade. Cum r este coeficientul de actualizare pe o perioadă, rezultă egalitatea pe care am scris-o.

Se vede clar de aici că o politică optimală în sensul valorii actualizate (minime sau maxime) depinde esențial de coeficientul de actualizare r, lucru care iarăși are o legătură strînsă cu logica economico-financiară. Așa de exemplu într-o situație economică în care intervine o dobîndă mică va fi preferată o politică de scadență îndelungată, în timp ce o dobîndă mare impune o scadență mai apropiată. Operația de actualizare are drept efect diminuarea exponențială a valorii, pe măsură ce perioadele cărora le este atașată se îndepărtează în viitor.

Și în acest caz este adevărată o teoremă de existență și unicitate care stă la baza aproximării valorii politicii optimale.

Scanned with OKEN Scanner

Teoremă: Fie un program dinamic staționar unde Γ_x este definită pentru orice $x \in \mathbf{X}$, $\Gamma_x \subset \mathbf{X}$, iar $v(x, x')$ este definită pentru orice $x \in \mathbf{X}$ și orice $x' \in \Gamma x$.

Dacă

1) $v(x, x')$ este uniform mărginită pentru orice x astfel încît :

$$x \in \mathbf{X}, \ |x| \leqslant c, \quad x' \in \Gamma x$$

avem :

$$|v(x, x')| \leqslant m(c)$$

2) $x' \in \Gamma x$ și $|x'| < |x|$ oricare ar fi $x \in \mathbf{X}$ atunci ecuația funcțională

$$f(x, r) = r \operatorname*{opt}_{x' \in \Gamma x} [v(x, x') + f(x', r)]$$

unde $0 \leqslant r < 1$ are o soluție unică care este uniform mărginită pentru orice x astfel încît $|x| \leqslant c$.

Șirul $\{f_n(x, r)\}$ definit prin relațiile

$$f_0(x, r) = 0$$

$$f_n(x, r) = r \operatorname*{opt}_{x' \in \Gamma x} [v(x, x') + f_{n-1}(x', r)]$$

converge către $f(x, r)$.

În plus dacă $v(x, x')$ este continuă, atunci $f(x, r)$ este continuă.

6.8. Criteriul valorii medii pe o perioadă

Așa, după cum am menționat, actualizarea valorilor este justificată ori de cîte ori este vorba de o valoare monetară sau de un echivalent, ceea ce nu se întîmplă întotdeauna în practică.

În situațiile cînd nu apare o valoare se utilizează adesea un alt criteriu de apreciere a politicilor și anume criteriul valorii medii pe o perioadă.

Pentru a preciza lucrurile vom raționa pentru început pe un program dinamic cu orizont limitat (cu N perioade).

Dacă $\pi = (x_0, x_1, \ldots, x_N)$ este o politică arbitrară și $s_N = \sum_{n=1}^{N} v_n(x_{n-1}, x_n)$ valoarea acestei politici, atunci prin definiție se numește valoare medie pe perioadă expresia :

$$\sigma_N = \frac{1}{N} \sum_{n=1}^{N} v_n(x_{n-1}, x_n) = \frac{s_N}{N}.$$

În cazul unui număr finit de perioade, a compara două politici este același lucru cu a compara valorile lor totale sau valorile medii pe o

Scanned with OKEN Scanner

perioadă, deci în cazul acesta criteriul valorii totale şi cel al valorii medii pe o perioadă sînt echivalente.

Dacă notăm cu $\varphi_{N|}(x)$ valoarea medie pe o perioadă, optimală pentru o stare iniţială x arbitrară, atunci :

$$\varphi_N(x) = \frac{1}{N} f_N(x)$$

Utilizînd acum relaţiile de recurenţă (evidenţiate pentru un program dinamic staţionar)

$$f_n(x) = \operatorname*{opt}_{x' \in \Gamma x} [v(x, x') + f_{n-1}(x')]$$

obţinem imediat :

$$\varphi_n(x) = \operatorname*{opt}_{x' \in \Gamma x} \left[\frac{v(x, x')}{n} + \frac{n-1}{n} \varphi_{n-1}(x') \right]$$

Dacă acum presupunem că şirul $\{\varphi_n(x)\}$ are limită cînd $n \to \infty$, limita fiind $\varphi(x)$ care reprezintă valoarea medie pe o perioadă a unei politici optimale pe o infinitate de perioade atunci obţinem ecuaţia funcţională pe care o satisface $\varphi(x)$:

$$\varphi(x) = \operatorname*{opt}_{x' \in \Gamma x} [\varphi(x')]$$

Această ecuaţie funcţională impune anumite condiţii asupra funcţiei $\varphi(x)$ însă nu o determină complet, dat fiind faptul că valoarea $v(x, x')$ nu intervine în această ecuaţie funcţională.

Cu toate acestea se poate aplica metoda aproximaţiilor succesive utilizînd relaţia :

$$\varphi_n(x) = \operatorname*{opt}_{x' \in \Gamma x} \left[\frac{v(x, x')}{n} + \frac{n-1}{n} \varphi_{n-1}(x') \right].$$

Scanned with OKEN Scanner

Capitolul VII

ELEMENTE DE TEORIA AȘTEPTĂRII

7.1. Noțiuni introductive

În procesele social-economice apare adesea un consum de timp inutil și nedorit al mijloacelor de producție, resurselor sau oamenilor, care se datorește imposibilității de a corela temporal diverse activități care se intercondiționează. Iau naștere astfel fenomenele de așteptare.

Exemple de fenomene de așteptare: utilaje care așteaptă nefolosite din lipsă de materiale pe un șantier sau într-o fabrică; cozile care le efectuează cumpărătorii la magazine sau la unități de servire; așteptarea tonului sau a numărului liber la telefon; un tren de marfă care nu se descarcă la timp datorită numărului insuficient de descărcători etc.

Un fenomen de așteptare se caracterizează prin trei elemente principale: 1) *consumatorii*, care solicită un serviciu, 2) *șirul de așteptare* sau coada care se formează în cazul în care consumatorii trebuie să aștepte și 3) *stația de servire*, a cărei menire este satisfacerea cererilor.

Vom introduce cîteva noțiuni fundamentale din teoria așteptării pe baza unui exemplu concret. Să presupunem că la o lucrare de terasamente s-au repartizat 3 excavatoare și o coloană de 15 autobasculante. Consumatorii, adică autobasculantele, se prezintă în fața stației de servire care este alcătuită din 3 posturi (excavatoarele); în cazul în care la stația de servire sînt prezente la un moment dat, mai mult decît trei autobasculante se formează un șir de așteptare. Consumatorii se așază în șirul de așteptare după o anumită regulă sau disciplină, care, în cazul nostru, este „în ordinea sosirii" sau „primul sosit, primul servit". Consumatorii sosesc după o anumită lege, cu alte cuvinte au un ritm de sosire; în mod corespunzător există și un ritm de servire.

7.2. Elemente de bază într-un model de așteptare

1. *Venirile (sosirile) în sistem*, se presupun în general ca fiind întîmplătoare și independente (există însă și situații — deci și modele în care venirile nu mai sînt independente).

Scanned with OKEN Scanner

Presupunem de asemenea, că venirile sînt poissoniene, adică numărul de veniri într-o unitate de timp este o variabilă aleatoare repartizată Poisson. În acest caz, se demonstrează că intervalul dintre două veniri succesive este o variabilă aleatoare repartizată exponențial negativ.

Mai trebuie făcută precizarea că poate fi vorba de veniri dintr-o populație finită sau infinită.

2. *Durata serviciului (timpul de servire)* este de asemenea o variabilă aleatoare care de obicei se presupune că este independentă de la o servire la alta și independentă de veniri.

Fiind o variabilă aleatoare se presupune că are o lege de repartiție determinată care poate fi de exemplu exponențial negativă, uniformă pe un interval $[a, b]$, χ^2, Gama, Erlang etc.

În fiecare model concret se va specifica repartiția duratei serviciului.

3. *Factorul de serviciu* sau intensitatea de trafic se definește ca raportul $\rho = \frac{\lambda}{\mu}$ care indică raportul între numărul mediu de unități ce sosesc în sistem și numărul mediu de unități ce pleacă din sistem într-o unitate de timp aleasă. Factorul de serviciu are o deosebită importanță în teoria firelor de așteptare, deoarece odată stabilite repartițiile venirilor și plecărilor, toate caracteristicile numerice ale modelului de așteptare studiat se pot exprima cu ajutorul acestui indicator. El indică imediat dacă este vorba de un sistem cu aglomerare nelimitată sau nu, după cum $\rho \geqslant 1$ sau $\rho < 1$.

4. *Stațiile de serviciu.* Prin stație de serviciu se înțelege mecanismul care deservește cererea formulată de unități.

Rolul stației îl joacă în aplicațiile practice unul sau mai mulți oameni una sau mai multe mașini, culoarele de mers într-o intersecție etc.

Există modele cu o singură stație de serviciu, sau cu mai multe stații. În cazul modelelor cu mai multe stații, ele pot fi dispuse în serie într-o ordine determinată și unitățile sînt obligate să treacă prin fiecare stație, sau pot fi dispuse în paralel, fiecare jucînd același rol în satisfacerea cererilor formulate de unități noi; ne vom referi la acest din urmă caz cînd vom considera mai multe stații.

5. *Disciplina sistemului* este un ansamblu de reguli prin care se precizează modul în care se vor dispune unitățile în șirul de așteptare precum și ordinea în care ele vor fi servite.

Disciplina cea mai simplă și în același timp naturală, este disciplina care precizează pe scurt, „primul venit, primul servit".

Pot exista și alte discipline cum ar fi de exemplu cazul modelelor în care unele unități au prioritate la servire față de alte unități sau cazul modelelor cu unități nedisciplinate.

6. *Modul de servire a unităților în sistem.* În funcție de model vom întîlni servicii individuale în care o unitate este servită o singură dată și apoi unitatea părăsește definitiv sistemul. Stația eliberată de unitatea ce a fost servită rămîne inactivă pînă la sosirea altei unități. Pot exista însă și alte tipuri de servire :

a) Servicii în bloc, în care sînt servite toate unitățile ce așteaptă pînă la un număr stabilit dinainte.

Cînd stația a terminat de servit un bloc servește al doilea bloc ș.a.m.d.

b) Servicii în masă, în care stația lucrează în continuu pe anumite intervale date de timp.

Scanned with OKEN Scanner

În fiecare interval de timp sînt servite toate unităţile care pînă la sfîrşitul intervalului precedent se găseau în aşteptare sau care ajung în acea perioadă de timp, numărul de unităţi ce intră în staţie neputînd depăşi un număr dat.

În vederea efectuării analizei unui model de aşteptare este necesar să cunoaştem o serie de caracteristici ale modelului.

Printre caracteristicile cele mai importante enumerăm:

— numărul mediu de unităţi în sistem (în şirul de aşteptare şi în curs de servire);
— numărul mediu de unităţi în curs de servire;
— numărul mediu de unităţi în şirul de aşteptare;
— numărul mediu de staţii neocupate;
— numărul mediu de unităţi ce sosesc într-o unitate dată de timp;
— timpul mediu de servire;
— timpul mediu de aşteptare în sistem;
— timpul mediu de aşteptare în şir etc.

Acestea sînt numai unele dintre caracteristicile numerice pe care vom căuta să le punem în evidenţă în cadrul modelelor ce le vom aborda. Precizăm faptul că spaţiul limitat nu ne permite să abordăm decît modele cu una sau mai multe staţii în care venirile sînt Poissoniene de parametru λ iar durata serviciilor este o variabilă aleatoare repartizată exponenţial negativ de parametru μ.

În cele ce urmează vom trata mai întîi modele cu o singură staţie de servire, apoi vom considera cazul modelelor cu mai multe staţii.

7.3. Modele de aşteptare cu o singură staţie de serviciu

Ne vom ocupa mai întîi de cazul unui model cu o singură staţie cu veniri dintr-o populaţie infinită atît în ipoteza posibilităţii de formare a unui şir de aşteptare nelimitat cît şi în ipoteza unui şir de aşteptare limitat.

7.3.1. Model cu o singură staţie de servire cu veniri dintr-o populaţie infinită şi cu şir de aşteptare nelimitat

Pentru a pune în evidenţă ecuaţiile modelului vom preciza ipotezele în care ne situăm:

a) Venirile în sistem sînt infinite, întîmplătoare, independente unele de altele şi numărul de veniri pe unitatea de timp este o variabilă aleatoare repartizată Poisson de parametru λ.

b) Serviciile sînt independente între ele şi independente de veniri iar durata serviciului este o variabilă aleatoare repartizată exponenţial negativ de parametru μ. O unitate odată servită părăseşte sistemul definitiv.

c) Există o singură staţie ce asigură serviciul solicitat de către unităţi.

d) Disciplina sistemului este cea obişnuită şi anume: primul venit este primul servit (unităţile intră în staţia de serviciu în ordinea venirii lor

Scanned with OKEN Scanner

în sistem, neexistînd unităţi nedisciplinate şi nici unităţi care să aibă prioritate la servire faţă de alte unităţi).

În aceste ipoteze de funcţionare a sistemului, vom analiza modelul plecînd de la momentul iniţial $t = 0$.

Dacă notăm cu $\delta > 0$ momentul cînd înregistrăm o venire şi cu $\mathbf{A}(\delta)$ probabilitatea de a avea o venire la momentul $\delta > 0$, plecînd de la starea iniţială $t = 0$ atunci în ipotezele pe care le-am fixat avem * :

$$\mathbf{A}(\delta) = \lambda \int_0^{\delta} e^{-\lambda x}\, dx = 1 - e^{-\lambda\delta}.$$

Dezvoltînd în serie în jurul originii, funcţia $e^{-\lambda\delta}$ se obţine :

$$\mathbf{A}(\delta) = 1 - \left[1 - \lambda\delta + \frac{(\lambda\delta)^2}{2!} + \ldots\right] = \lambda\,\delta + \mathbf{O}(\delta),$$

unde $\mathbf{O}(\delta)$ are proprietatea că $\lim\limits_{\delta \to 0} \dfrac{\mathbf{O}(\delta)}{\delta} = 0.$

Dacă notăm cu $\mathbf{A}(t,\ t + \delta)$ probabilitatea de a înregistra o venire în intervalul de timp $(t,\ t + \delta)$ atunci în ipotezele stabilite obţinem :

$$\mathbf{A}(t,\ t + \delta) = \lambda \int_t^{t+\delta} e^{-\lambda x}\, dx = e^{-\lambda t} - e^{-(t+\delta)\lambda} = e^{-\lambda t}\,(1 - e^{-\lambda\delta}),$$

adică :

$$\mathbf{A}(t,\ t + \delta) = [1 - \mathbf{A}(t)]\ \mathbf{A}(\delta).$$

(sau exprimat în cuvinte, probabilitatea de a înregistra o venire în intervalul de timp $(t,\ t + \delta)$ este egală cu probabilitatea a nu avea nici o venire la momentul t înmulţită cu probabilitatea de a înregistra o venire la momentul δ).

În mod analog se calculează probabilitatea ca o unitate să părăsească sistemul la un moment dat $\delta > 0$ şi pe care o notăm $\mathbf{B}(\delta)$:

$$\mathbf{B}(\delta) = \mu\delta + \mathbf{O}(\delta).$$

Ipotezele formulate ne permit să exprimăm probabilitatea ca la momentul $t + \delta$ să avem n unităţi în sistem şi pe care o notăm $\mathbf{P}_n(t + \delta)$:

$$\mathbf{P}_n(t + \delta) = \mathbf{P}_{n-1}(t)\ \mathbf{A}(\delta)\ [1 - \mathbf{B}(\delta)] + \mathbf{P}_n(t)\ [1 - \mathbf{A}(\delta)]\ [1 - \mathbf{B}(\delta)] + \\ + \mathbf{P}_{n+1}(t)\ \ [1 - \mathbf{A}(\delta)]\ \ \mathbf{B}(\delta) + 0(\delta).$$

---

* Din ipoteza că venirile sînt Poissoniene rezultă că timpul dintre două veniri succesive în sistem este o variabilă aleatoare repartizată exponenţial negativ de parametru λ, adică are densitatea de repartiţie :

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0 \quad , & x \leqslant 0 \end{cases} \quad \text{cu } \lambda > 0.$$

Scanned with OKEN Scanner

deoarece probabilitatea ca să fim în starea S_n (să existe n unităţi în sistem la momentul $t + \delta$) la momentul $t + \delta$ este suma următoarelor probabilităţi :

— probabilitatea ca la momentul t să fim în starea S_{n-1} şi în intervalul de timp de la t, la $t + \delta$ să vină o unitate şi să nu plece nici o unitate :

$$P_{n-1}(t)\, A(\delta)\, [1 - B(\delta)]$$

— probabilitatea ca la momentul t să fim în starea S_n şi în intervalul de timp de la t la $t + \delta$ să nu vină nici o unitate în sistem şi să nu plece nici o unitate :

$$P_n(t)\, [1 - A(\delta)]\, [1 - B(\delta)].$$

— probabilitatea ca la momentul t să fim în starea S_{n+1} şi în intervalul de timp de la t la $t + \delta$ să nu vină nici o unitate şi să plece o unitate :

$$P_{n+1}(t)\, [1 - A(\delta)]\, B(\delta)$$

— probabilitatea ca în intervalul de timp de la t la $t + \delta$ numărul unităţilor din sistem să varieze cu mai mult de unu : $O(\delta)$.

Înlocuind pe $A(\delta)$ şi $B(\delta)$ obţinem :

$$\begin{aligned} P_n(t + \delta) = {} & P_{n-1}(t)\, [\lambda\delta + O(\delta)]\, [1 - \mu\delta - O(\delta)] + \\ & + P_n(t)\, [1 - \lambda\delta - O(\delta)]\, [1 - \mu\delta - O((\delta)] + \\ & + P_{n+1}(t)\, [1 - \lambda\delta - O(\delta)]\, [\mu\delta + O(\delta)] + O(\delta), \end{aligned}$$

sau ordonînd după puterile lui δ :

$$\begin{aligned} P_n(t + \delta) = {} & P_{n-1}(t)\, \lambda\delta + P_n(t)\, [1 - (\lambda + \mu)\, \delta] + \\ & + P_{n+1}(t)\mu\delta + O(\delta). \end{aligned}$$

De aici se obţine :

$$\frac{P_n(t+\delta) - P_n(t)}{\delta} = \lambda P_{n-1}(t) - (\lambda + \mu)\, P_n(t) + \mu P_{n+1}(t) + \frac{O(\delta)}{\delta}.$$

Făcînd pe δ să tindă la zero, membrul al doilea are limită, deci şi membrul întîi are limită şi obţinem :

$$\frac{d P_n(t)}{dt} = \lambda P_{n-1}(t) - (\lambda + \mu) P_n(t) + \mu\, P_{n+1}(t).$$

$$n = 1, 2, 3, \ldots$$

Pentru $n = 0$ obţinem o ecuaţie specială dacă exprimăm $P_0(t + \delta)$:

$$P_0(t + \delta) = P_0(t)\, [1 - A(\delta)] + P_1(t)\, [1 - A(\delta)]\, B(\delta) + O(\delta)$$

de unde înlocuind $A(\delta)$ şi $B(\delta)$, trecînd pe $P_0(t)$ în membrul întîi şi împărţind cu δ obţinem :

$$\frac{P_0(t+\delta) - P_0(t)}{\delta} = -\lambda\, P_0(t) + \mu\, P_1(t) + \frac{O(\delta)}{\delta}.$$

Scanned with OKEN Scanner

De aici făcînd pe δ să tindă la zero rezultă :

$$\frac{d\,P_0(t)}{dt} = -\lambda\,P_0(t) + \mu\,P_1(t).$$

Prin urmare modelul considerat de noi este caracterizat de sistemul de ecuaţii diferenţiale :

$$\frac{d\,P_n(t)}{dt} = \lambda\,P_{n-1}(t) - (\lambda+\mu)\;P_n(t) + \mu\,P_{n+1}(t) \qquad n = 1,\ 2,\ 3,\ldots$$

$$\frac{d\,P_0(t)}{dt} = -\lambda\,P_0(t) + \mu\,P_1(t).$$

Rezolvarea acestui sistem de ecuaţii diferenţiale ne permite să aflăm probabilitatea $P_n(t)$ ca la momentul t să avem în sistem n unităţi. Nu vom determina aceste probabilităţi în cazul general ci în cazul regimului staţionar de funcţionare (sau ale echilibrului statistic) caracterizat de faptul că :

$$\lim_{t\to\infty} \frac{d\,P_n(t)}{dt} = 0, \qquad n = 0,\ 1,\ 2,\ldots$$

În această situaţie $P_n(t) = p_n$, $n = 0,\ 1,\ 2,\ 3\ldots$ şi sistemul de ecuaţii diferenţiale se reduce la un sistem de ecuaţii algebrice liniare :

$$\begin{cases} \lambda\,p_{n-1} - (\lambda+\mu)\;p_n + \mu\,p_{n+1} = 0, & n = 1,\ 2,\ 3,\ldots \\ -\lambda p_0 + \mu\,p_1 = 0 \end{cases}$$

care se rezolvă uşor.

Pentru aceasta relaţia valabilă pentru $n=1,2,\ldots$ o scriem sub forma :

$$\lambda\,p_{n-1} - \mu p_n = \lambda p_n - \mu\,p_{n+1}, \qquad n = 1,\ 2,\ 3,\ldots$$

Notînd :

$$z_n = \lambda\,p_{n-1} - \mu\,p_n$$

avem

$$z_n = z_{n+1}, \quad n = 1,\ 2,\ 3,\ldots$$

Cum $z_1 = \lambda\,p_0 - \mu p_1 = 0$ rezultă $z_n = 0 \quad n = 1,\ 2,\ldots$

şi de aici :

$$p_n = \frac{\lambda}{\mu}\,p_{n-1} = \rho\,p_{n-1}, \quad n = 1,\ 2,\ldots$$

adică :

$$p_n = \rho^n p_0, \quad n = 1,\ 2,\ldots$$

Să determinăm probabilitatea p_0. Dacă $\rho < 1$ atunci :

$$\sum_{n=0}^{\infty} \rho^n = \frac{1}{1-\rho} \text{ şi cum } \sum_{n=0}^{\infty} p_n = 1 \text{ rezultă}$$

$$1 = p_0 \sum_{n=0}^{\infty} \rho^n = p_0 \frac{1}{1-\rho},$$

Scanned with OKEN Scanner

și de aici $p_0 = 1 - \rho$ iar cu aceasta

$$p_n = \rho^n (1 - \rho) \quad n = 0, 1, 2, \ldots$$

Să vedem mai întîi care este valoarea maximă a probabilității $p_n(\rho)$. Pentru aceasta rezolvăm ecuația :

$$\frac{d\, p_n(\rho)}{d\rho} = 0 \quad \text{adică} \quad \rho^{n-1} [n - \rho\ (n+1)] = 0$$

care ne conduce la soluția $\rho = \frac{n}{n+1}$ și deci

$$\max_{\rho} p_n(\rho) = \left(\frac{n}{n+1}\right)^n \cdot \frac{1}{n+1}$$

În calculele economice se utilizează o serie de parametri, pe care-i vom deduce în cele ce urmează.

Să notăm cu **Z** variabila aleatoare ce ne dă numărul de unități în sistem. Atunci Z are repartiția :

$$\mathbf{Z} : \begin{pmatrix} 0 & 1 & 2 & \ldots & n & \ldots \\ p_0 & p_1 & p_2 & \cdots & p_n & \ldots \end{pmatrix}$$

De aici rezultă imediat probabilitatea ca numărul unităților din sistem să nu depășească un număr n :

$$\mathbf{P}(\mathbf{Z} \leqslant n) = \sum_{j=1}^{n} \mathbf{P}(\mathbf{Z} = j) = \sum_{j=1}^{n} p_j = (1 - \rho) \sum_{j=1}^{n} \rho^j = 1 - \rho^{n+1},$$

și totodată $\mathbf{P}(\mathbf{Z} > n) = \rho^{n+1}$.

În particular $\mathbf{P}(\mathbf{Z} > 0) = \rho$.

$\mathbf{P}(\mathbf{Z} > 0)$ reprezintă probabilitatea ca stația să fie ocupată și constatăm că această probabilitate este egală cu factorul de serviciu (intensitatea traficului).

Numărul mediu de unități în sistem :

$$\overline{\mathbf{Z}} = \mathbf{M}(\mathbf{Z}) = \sum_{n=0}^{\infty} n\, p_n = \sum_{n=0}^{\infty} n(1 - \rho)\, \rho^n = \frac{\rho}{1-\rho}.$$

Numărul mediu de unități în șirul de așteptare.

Dacă notăm cu Q numărul de unități în șirul de așteptare, atunci :

$$\mathbf{Q} : \begin{pmatrix} 0 & 1 & 2 \ldots n & \ldots \\ p_0 + p_1 & p_2 & p_3 \ldots p_{n+1} & \ldots \end{pmatrix}$$

Scanned with OKEN Scanner

Deci aici rezultă:

$$\overline{Q} = M(Q) = \sum_{n=1}^{\infty} (n-1)\, p_n = (1-\rho)\ \rho^2\ (1 + 2\ \rho + 3\ \rho^2 + \ldots) = \frac{\rho^2}{1-\rho}.$$

Această caracteristică se mai poate calcula și altfel și anume:

Fie S numărul de unități în curs de servire. Atunci

$$Z = S + Q$$

Însă

$$S : \begin{pmatrix} 0 & 1 \\ p_0 & 1-p_0 \end{pmatrix}$$

deci

$$\overline{S} = M(S) = 1 - p_0 = \rho$$

și deci

$$\overline{Q} = \overline{Z} - \overline{S} = \frac{\rho}{1-\rho} - \rho = \frac{\rho^2}{1-\rho^2}.$$

Se constată imediat că

$$\overline{Q} = \rho\,\overline{Z} = \frac{\lambda}{\mu}\,\overline{Z},$$

adică:

$$\frac{\overline{Q}}{\lambda} = \frac{\overline{Z}}{\mu}$$

În problemele practice intervin adesea situații cînd ne interesează timpul mediu de așteptare în șir (pe care-l notăm cu W) sau timpul mediu de așteptare în sistem (pe care-l notăm cu W*).

Timpul mediu de așteptare în șir se calculează în modul următor:

$W = M$ [(numărul de unități în sistem) × (timpul de servire al unei unități)]

$= M$ (numărul de unități în sistem) × M (timpul de servire al unei unități)

$$= \overline{Z} \cdot \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}.$$

Rezultă acum imediat timpul mediu de așteptare în sistem și anume:

$$W^* = W + \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)} + \frac{1}{\mu} = \frac{1}{\mu} \cdot \frac{1}{1-\rho}.$$

Scanned with OKEN Scanner

Se poate calcula imediat funcţia generatoare de momente pentru variabila aleatoare $\mathbf{Z}$:

$$\psi(t) = \mathbf{M}(e^{t\mathbf{Z}}) = \sum_{n=0}^{\infty} e^{tn}\, p_n = (1-\rho) \sum_{n=0}^{\infty} (e^{t+\ln\rho})^n =$$

$$= \frac{1-\rho}{1-e^{t+\ln\rho}}$$

De aici urmează

$$\mathbf{M}(\mathbf{Z}^2) = \frac{\rho+\rho^2}{(1-\rho)^2}.$$

şi deci

$$\mathbf{D}^2(\mathbf{Z}) = \mathbf{M}(\mathbf{Z})^2 - \mathbf{M}^2(\mathbf{Z}) = \frac{\rho}{(1-\rho)^2}.$$

Se constată imediat că $\mathbf{D}^2(\mathbf{Z})$ este funcţie crescătoare de ρ şi deci obţinem o precizie cît mai bună în cazul modelelor în care ρ este cît mai apropiat de zero.

Un indicator frecvent utilizat în analiza sistemelor de aşteptare este indicatorul $P(>\tau)$, care reprezintă probabilitatea ca o unitate să aştepte în şir un timp superior unui timp dat τ.

Pentru a determina această probabilitate vom determina mai întîi densitatea de repartiţie a timpului de aşteptare în şir.

Vom considera un moment t_0 şi vom urmări ce se întîmplă cu o unitate reperată la acest moment. Să calculăm probabilitatea elementară $f(\tau)\mathrm{d}\tau$ ca unitatea reperată să aştepte un timp cuprins între τ şi $\tau + \mathrm{d}\tau$.

Dacă la momentul t_0 cînd unitatea specificată a fost reperată ca intrînd în sistem, înaintea ei se află alte n unităţi, atunci probabilitatea elementară menţionată se obţine ca suma probabilităţilor $p(n, \tau)\ \mathrm{d}\tau$, unde $p(n,\tau)\ \mathrm{d}\tau$ reprezintă probabilitatea elementară ca la momentul t_0 să avem în sistem n unităţi, iar unitatea reperată să aştepte un timp cuprins între τ şi $\tau + \mathrm{d}\tau$.

Cum timpul de serviciu urmează o lege exponenţial negativă de parametru μ, numărul de plecări urmează o lege Poisson, atîta timp cît staţia este ocupată.

Aşadar dacă staţia este ocupată, probabilitatea ca să devină disponibilă în intervalul de timp de lungime Δt este $\mu\,\Delta t$ şi deci repartiţia numărului de plecări atîta timp cît staţia este ocupată este dată de :

$$p(s) = \frac{(\mu t)^s e^{-\mu t}}{s!}, \quad s = 0,\ 1,\ 2, \ldots$$

Dacă $n = 0$, atunci $p(0, \tau) = 0$ ceea ce rezultă imediat din semnificaţia probabilităţii $p(n, \tau)$.

Scanned with OKEN Scanner

Presupunem $n > 0$ și atunci :

$$p(n, \tau)\, d\tau = p_n \cdot \mathbf{P}\left(\begin{array}{l}\text{pleacă } s = n-1 \text{ unități servite în intervalul de timp } \tau\end{array} \middle/ \begin{array}{l}\text{În sistem există } n \text{ unități la momentul } t_0\end{array}\right) \times$$

$$\times\, \mathbf{P}\left(\begin{array}{l}\text{să plece o unitate în intervalul de timp } d\tau\end{array} \middle/ \begin{array}{l}\text{La începutul intervalului } \tau \text{ avem } n \text{ unități; după scurgerea intervalului de timp } \tau \text{ au plecat } s = n-1 \text{ unități)}\end{array}\right)$$

Cum am presupus că sistemul este în regim staționar avem :

$$p_n = (1 - \rho)\rho^n.$$

Pentru cele două probabilități condiționate putem scrie :

$$\mathbf{P}\left(\begin{array}{l}\text{Pleacă } s = n-1 \text{ unități servite în intervalul de timp } \tau\end{array} \middle/ \begin{array}{l}\text{În sistem există } n \text{ unități la momentul } t\end{array}\right) = \frac{(\mu\tau)^{n-1} e^{-\mu\tau}}{(n-1)!}$$

$$\mathbf{P}\left(\begin{array}{l}\text{Să plece o unitate din sistem în intervalul } d\tau\end{array} \middle/ \begin{array}{l}\text{La începutul intervalului } \tau \text{ erau } n \text{ unități; după scurgerea intervalului } \tau \text{ au plecat } s = n-1 \text{ unități}\end{array}\right) = \mu d\tau$$

Deci :

$$p(n, \tau)\, d\tau = \rho^n(1 - \rho)\frac{(\mu\tau)^{n-1}e^{-\mu\tau}}{(n-1)!}\mu\, d\tau, \quad n = 1, 2, \ldots$$

De aici urmează că :

$$f(\tau)\, d\tau = \sum_{n=1}^{\infty} p(n, \tau)\, d\tau = \sum_{n=1}^{\infty} \rho^n(1 - \rho)\frac{(\mu\tau)^{n-1} e^{-\mu\tau}}{(n-1)!}\mu\, d\tau =$$

$$= (1 - \rho)\, \rho\, \mu\, e^{-\mu\tau(1-\rho)}\, d\tau.$$

Urmează acum că

$$\mathbf{P}(> \tau) = \int_{\tau}^{\infty} f(t)\, dt = (1 - \rho)\, \rho\mu \int_{\tau}^{\infty} e^{-\mu t(1-\rho)}\, dt = \rho\, e^{-\tau\mu(1-\rho)}.$$

În particular probabilitatea ca o unitate să aștepte este $\mathbf{P}(> 0) = \rho$ și de aici probabilitatea ca unitatea să nu aștepte (sau altfel spus probabilitatea ca stația să nu fie ocupată) este :

$$1 - \mathbf{P}(> 0) = 1 - \rho.$$

sau

$$\mathbf{P}(Z = 0) = p_0 = 1 - \rho.$$

Scanned with OKEN Scanner

și în felul acesta am ajuns (pe altă cale) la rezultatul pe care-l obținusem deja.

Exemplul 7.1. La un ghișeu CEC se prezintă pentru a efectua operațiuni CEC în medie 64 persoane pe zi. Ghișeul este deschis 8 ore pe zi iar operatorul de la ghișeu servește în medie 12 persoane pe oră. Dintr-o verificare statistică s-a constatat că venirile sînt poissoniene iar durata serviciului este o variabilă aleatoare exponențială negativă cu parametrul μ.

Să se calculeze: probabilitățile p_n ca în sistem să se afle n persoane, probabilitatea ca la un moment dat să se afle cel mult 4 solicitanți, probabilitatea ca în sistem să se afle cel puțin 5 solicitanți,

— numărul mediu de solicitanți existenți la CEC,

— numărul mediu de solicitanți care așteaptă la rînd pentru a fi serviți;

— media de inactivitate a stației de servire.

— timpul mediu de așteptare al unui solicitant;

— probabilitatea ca un solicitant oarecare să aștepte mai mult de 15 minute.

Soluție — Utilizăm un model cu o singură stație cu veniri dintr-o populație infinită, veniri poissoniene durata serviciului exponențial negativ și care funcționează în regim staționar.

Atunci:

$$p_n = \rho^n p_0 \text{ cu } p_0 = 1 - \rho, \ \rho = \frac{\lambda}{\mu}.$$

Luînd unitatea de timp ora, avem

$$\rho = \frac{\frac{64}{8}}{12} = \frac{8}{12} = \frac{2}{3}$$

$$p_0 = 1 - \frac{2}{3} = \frac{1}{3}, \quad p_n = \left(\frac{2}{3}\right)^n \cdot \frac{1}{3}, \quad n = 1, 2, 3, \ldots \quad p_n = \frac{2}{3} p_{n-1},$$

de exemplu

$$p_1 = \frac{2}{9}, \quad p_2 = \frac{4}{27}, \quad p_3 = \frac{8}{81}, \ldots$$

Dacă Z este numărul de unități în sistem, atunci:

$$\mathbf{P}(Z \leqslant 4) = \sum_{n=0}^{4} p_n = \frac{1}{3} + \frac{2}{9} + \frac{4}{27} + \frac{8}{81} + \frac{16}{243} = \frac{211}{243}$$

$$\mathbf{P}(Z \geqslant 5) = 1 - \mathbf{P}(Z \leqslant 4) = 1 - \frac{211}{243} = \frac{32}{243} \approx 0{,}13$$

Scanned with OKEN Scanner

ceea ce înseamnă că în medie în 13 % din cazuri așteaptă să li se efectueze serviciul, cel puțin 5 persoane

$$\bar{Z} = M(Z) = \frac{\rho}{1-\rho} = \frac{\frac{2}{3}}{1-\frac{2}{3}} = 2.$$

În medie la ghișeu sînt prezente 2 persoane.
Numărul mediu de persoane care așteaptă la rînd să fie servite :

$$\bar{Q} = M(Q) = \frac{\rho^2}{1-\rho} = \frac{4}{3},$$

adică în medie între una și două persoane.
Inactivitatea medie a stației :

$$\bar{S} = M(S) = 1 - \rho = \frac{1}{3}.$$

În medie a treia parte de timp stația este inactivă.
Timpul mediu de așteptare în sistem a unui solicitant :

$$W = \frac{\rho}{\mu(1-\rho)} = \frac{\lambda}{\mu(\mu-\lambda)} = \frac{8}{12(12-8)} = \frac{2}{3 \cdot 4} = \frac{1}{6}$$

deci un solicitant așteaptă în medie 10 minute.
Probabilitatea ca un solicitant să aștepte mai mult de 15 minute :

$$P\left(> \frac{1}{4}\right) = \rho\, e^{-\frac{1}{4}\mu(1-\rho)} = \frac{2}{3} e^{-\frac{1}{4} 12 \left(1-\frac{2}{3}\right)} = \frac{2}{3} e^{-1}.$$

Exemplul 7.2. Pe o stradă într-un anumit punct este instalată o cabină telefonică cu un singur aparat. Din măsurătorile statistice efectuate s-a constatat că numărul de sosiri ale solicitanților pe unitatea de timp (ora) urmează o repartiție Poisson de medie 4. Durata medie a unei convorbiri telefonice este de 5 minute.

Se cere să se determine, utilizînd un model de așteptare cu o stație, cu veniri dintr-o populație infinită în ipoteza funcționării în regim staționar :

— Probabilitatea ca o persoană care sosește în fața cabinei și dorește să telefoneze să aibă de așteptat.

— Timpul mediu de așteptare în sistem și timpul mediu de așteptare în șir.

— Numărul mediu de persoane care așteaptă la rînd.

Scanned with OKEN Scanner

— Să se stabilească dacă este oportun să se instaleze un al doilea post telefonic alături de cel existent știind că această măsură este luată numai dacă solicitanții așteaptă în medie peste 15 minute. Cît trebuie să fie intensitatea venirilor pentru a se instala un nou post telefonic.

— Probabilitatea ca un solicitant să aștepte mai mult de 20 minute pînă ce poate vorbi la telefon.

Soluție. Din datele problemei rezultă că $\mu = 12$ și deci $\rho = \frac{4}{12} = \frac{1}{3}$.

Urmează că probabilitatea ca o persoană să aștepte este $P(>0) = \rho = \frac{1}{3}$ adică în medie 33% din cazuri se așteaptă pentru a se putea telefona.

Timpul mediu de așteptare în șir (la coadă):

$$W = \frac{\rho}{\mu(1-\rho)} = \frac{\frac{1}{3}}{12\left(1-\frac{1}{3}\right)} = \frac{1}{24},$$

deci timpul mediu de așteptare în șir este de 2,5 minute.

Timpul mediu de așteptare în sistem:

$$W^* = W + \frac{1}{\mu} = \frac{1}{24} + \frac{1}{12} = \frac{3}{24} = \frac{1}{8},$$

adică timpul mediu de așteptare în sistem este de 7,5 minute.

Numărul mediu de persoane care așteaptă la rînd:

$$\bar{Q} = \frac{\rho^2}{1-\rho} = \frac{\left(\frac{1}{3}\right)^2}{1-\frac{1}{3}} = \frac{1}{6}.$$

Numărul mediu de persoane în sistem:

$$\bar{Z} = \frac{\rho}{1-\rho} = \frac{\frac{1}{3}}{1-\frac{1}{3}} = 0{,}5.$$

Am văzut că timpul mediu de așteptare în sistem este de 7,5 minute. Cum serviciul de telefoane instalează un nou post numai cînd timpul de așteptare în sistem este de peste 15 minute, rezultă că postul telefonic actual este suficient.

Scanned with OKEN Scanner

Pentru a vedea cît trebuie să fie rata venirilor pentru a instala un nou post telefonic vom rezolva ecuaţia :

$$\frac{\rho}{\mu(1-\rho)} > \frac{1}{4}, \quad \text{sau} \quad \frac{\lambda}{\mu(\mu-\lambda)} > \frac{1}{4},$$

adică

$$\frac{\lambda}{12(12-\lambda)} > \frac{1}{4}$$

de unde rezultă $\lambda > 9$ sosiri pe oră.

Probabilitatea ca un solicitant să aştepte mai mult de 20 de minute pînă ce să poată vorbi la telefon este dată de

$$P\left(>\frac{1}{3}\right) = \rho\, e^{-\frac{1}{3}\mu(1-\rho)} = \frac{1}{3} e^{-\frac{1}{3}12\cdot\frac{2}{3}} = \frac{1}{3} e^{-\frac{8}{3}}.$$

7.3.2. Model de aşteptare cu o singură staţie, cu veniri dintr-o populaţie infinită şi cu şir de aşteptare limitat

Pentru a înţelege mai bine fenomenul care conduce la un astfel de model vom face apel la un exemplu.

Într-o localitate există o staţie PECO pentru alimentarea automobilelor, dotată cu o singură pompă (sau cu mai multe pompe deservite de o singură persoană) şi să admitem ca această staţie dispune de un teren care permite staţionarea numai a unui număr N de automobile (cel mult). Ca atare şirul de aşteptare poate avea lungimea maximă de $N - 1$ unităţi, o unitate fiind în staţia de servire.

Capacitatea unui astfel de sistem este N unităţi, iar automobilele care necesită o alimentare dar care nu mai găsesc loc în şirul de aşteptare sînt nevoite să meargă în altă parte.

Sîntem aşadar în cazul unui model cu o singură staţie de serviciu, cu veniri dintr-o populaţie infinită, iar numărul n al stărilor posibile ale sistemului este cuprins între 0 şi N.

Admitem că venirile sînt poissoniene, durata serviciului o variabilă repartizată exponenţial negativ de parametru μ. Păstrăm neschimbate de asemenea celelalte elemente întîlnite la modelul anterior, inclusiv disciplina staţiei.

Ca atare rămîn valabile ecuaţiile diferenţiale :

$$\frac{dP_0(t)}{dt} = -\lambda P_0(t) + \mu P_1(t)$$

$$\frac{dP_n(t)}{dt} = \lambda P_{n-1}(t) - (\lambda + \mu) P_n(t) + \mu P_{n+1}(t), \; n = 1, 2, \ldots, N-1,$$

și să obținem ecuația pentru cazul cînd toate unitățile sînt în sistem. Exprimăm pentru aceasta $P_N(t+\delta)$:

$$P_N(t+\delta) = P_{N-1}(t)\ A(\delta)\ [1 - B(\delta)] + P_N(t)\ [1 - B(\delta)] + O(\delta).$$

Înlocuind $A(\delta)$ și $B(\delta)$, trecînd pe $P_N(t)$ în membrul întîi și împărțind cu δ obținem:

$$\frac{P_N(t+\delta) - P_N(t)}{\delta} = \lambda P_{N-1}(t) - \mu P_N(t) + \frac{O(\delta)}{\delta}.$$

și de aici făcînd pe $\delta \to 0$ obținem ecuația diferențială:

$$\frac{dP_N(t)}{dt} = \lambda P_{N-1}(t) - \mu P_N(t).$$

Presupunem că sîntem în cazul regimului staționar, adică:

$$\lim_{t\to\infty} \frac{dP_n(t)}{dt} = 0,\ n = 0,\ 1,\ 2,\ldots,\ N,$$

și în acest caz sistemul de ecuații diferențiale devine un sistem de ecuații algebrice:

$$\begin{cases} -\lambda p_0 + \mu p_1 = 0 \\ \lambda p_{n-1} - (\lambda + \mu) p_n + \mu p_{n+1} = 0, \quad n = 1, 2, \ldots, N-1, \\ \lambda p_{N-1} - \mu p_N = 0 \end{cases}$$

Notînd

$$z_n = \lambda p_{n-1} - \mu p_n$$

se obține relația:

$$z_n = z_{n+1},\ n = 1, 2, \ldots, N-1 \text{ și cum } z_1 = 0 \text{ rezultă } z_n = 0,$$

deci

$$p_n = \rho p_{n-1} = \rho^n p_0,\ n = 1, 2, \ldots, N.$$

Probabilitatea p_0 se determină imediat din condiția

$$\sum_{n=0}^{N} p_n = 1 \text{ adică } p_0 \sum_{n=0}^{N} \rho^n = 1 \text{ de unde rezultă:}$$

$$p_0 = \frac{1-\rho}{1-\rho^{N+1}}$$

Scanned with OKEN Scanner

Caracteristici numerice ale modelului :

Dacă notăm cu **Z** variabila aleatoare numărul de unităţi în sistem, atunci

$$\mathbf{Z} = \begin{pmatrix} 0 & 1 & 2 & \dots & n & \dots & \mathrm{N} \\ p_0 & p_1 & p_2 & \dots & p_n & \dots & p_{\mathrm{N}} \end{pmatrix}$$

şi de aici numărul mediu de unităţi în sistem :

$$\bar{\mathbf{Z}} = \mathbf{M}(\mathbf{Z}) = \sum_{n=0}^{\mathrm{N}} n p_n = \frac{1-\rho}{1-\rho^{\mathrm{N}+1}} \rho \sum_{n=1}^{\mathrm{N}} n \rho^{n-1}$$

Efectuind calculele se obţine :

$$\bar{\mathbf{Z}} = \rho \frac{1-(\mathrm{N}+1)\rho^{\mathrm{N}}+\mathrm{N}\rho^{\mathrm{N}+1}}{(1-\rho)(1-\rho^{\mathrm{N}+1})}$$

Fie acum **S** numărul de unităţi în curs de servire. Fiind o singură staţie avem :

$$\mathbf{S} : \begin{pmatrix} 0 & 1 \\ p_0 & 1-p_0 \end{pmatrix}$$

şi de aici :

$$\bar{\mathbf{S}} = \mathbf{M}(\mathbf{S}) = 1 - p_0 = \rho \frac{1-\rho^{\mathrm{N}}}{1-\rho^{\mathrm{N}+1}}.$$

Putem obţine imediat acum numărul mediu **Q** de unităţi în şirul de aşteptare, deoarece

$$\mathbf{Q} = \mathbf{Z} - \mathbf{S}$$

Urmează că :

$$\bar{\mathbf{Q}} = \mathbf{M}(\mathbf{Q}) = \mathbf{M}(\mathbf{Z}) - \mathbf{M}(\mathbf{S}) = \rho \frac{1-(\mathrm{N}+1)\rho^{\mathrm{N}}+\mathrm{N}\rho^{\mathrm{N}+1}}{(1-\rho)(1-\rho^{\mathrm{N}+1})} - \rho \frac{1-\rho^{\mathrm{N}}}{1-\rho^{\mathrm{N}+1}} =$$

$$= \rho^2 \frac{1-\mathrm{N}\rho^{\mathrm{N}}+(\mathrm{N}-1)\rho^{\mathrm{N}}}{(1-\rho)(1-\rho^{\mathrm{N}+1})}$$

Putem calcula de asemenea dispersia variabilei aleatoare **Z** :

$$\sigma_{\mathbf{Z}}^2 = \sum_{n=0}^{\mathrm{N}} n^2 p_n - \bar{\mathbf{Z}}^2 = \frac{\rho-(\mathrm{N}+1)^2\rho^{\mathrm{N}+1}(1+\rho^2)+2\mathrm{N}^2\rho^{\mathrm{N}+2}-\rho^{2\mathrm{N}+2}}{(1-\rho)^2(1-\rho^{\mathrm{N}+1})^2}$$

Scanned with OKEN Scanner

Adesea pentru numărul mediu de unităţi în sistem se foloseşte o formulă aproximativă care se deduce relativ simplu din formula exactă :

$$\bar{Z} \simeq \begin{cases} \rho + \rho^2, & \text{dacă } \rho << 1 \\ \frac{N}{2} + \frac{1}{12} N(N+2)(\rho - 1), & \text{dacă } \rho \to 1 \\ N - \frac{1}{\rho}, & \text{dacă } \rho >> 1. \end{cases}$$

O formulă aproximativă asemănătoare se poate da şi pentru indicatorul $\bar{Q}$.

Exemplul 7.3. Într-o localitate există un centru de reparat televizoare, reparaţiile fiind asigurate de către un singur depanator. Un studiu statistic efectuat asupra venirilor şi duratelor de depanare a condus la concluzia că venirile sînt poissoniene cu $\lambda = 5$ sosiri pe zi (ziua de lucru este de 8 ore iar durata medie de reparare a unui televizor fiind de 90 minute.

Capacitatea de depozitare pentru televizoare în atelier fiind limitată la 15 bucăţi se cere să se calculeze :

— probabilitatea ca la un moment dat depanatorul să nu aibă de lucru ;

— probabilitatea ca în spaţiul de depozitare să se afle cel mult 5 televizoare.

— numărul mediu de televizoare existente în atelier.

— să se compare numărul mediu de unităţi în sistem în cazul cînd se foloseşte relaţia exactă şi relaţia aproximativă.

Soluţie: Sîntem în cazul unui model de aşteptare cu veniri dintr-o populaţie infinită cu şir limitat.

Luînd unitatea de timp ziua de lucru 8 ore avem

$$\lambda = 5, \quad \mu = \frac{16}{3}; \quad \rho = \frac{\lambda}{\mu} = \frac{15}{16}$$

Atunci

$$p_0 = \frac{1-\rho}{1-\rho^{16}} = \frac{1-\frac{15}{16}}{1-\left(\frac{15}{16}\right)^{16}} \simeq 0{,}096.$$

Probabilitatea ca să se afle cel mult 5 televizoare este :

$$P = p_0 + p_1 + p_2 + p_3 + p_4 + p_5 = 0{,}484.$$

Numărul mediu de unităţi în sistem (televizoare existente pentru reparaţii) :

$$\bar{Z} = 6{,}467$$

Cum ρ este apropiat de valoarea *1* folosim relaţia aproximativă :

$$\bar{Z} \simeq \frac{N}{2} + \frac{1}{12} N(N+2)(\rho - 1) = 6{,}172.$$

Scanned with OKEN Scanner

Se constată că s-au obținut valori suficient de apropiate.
În medie sînt aproximativ *6* televizoare în atelierul de reparații.

7.3.3. Model cu o singură stație și cu unitățile provenind dintr-o populație finită

Modelele de acest gen mai sînt cunoscute sub denumirea de modele închise. Situații concrete care se pot modela în felul acesta intervin foarte frecvent.

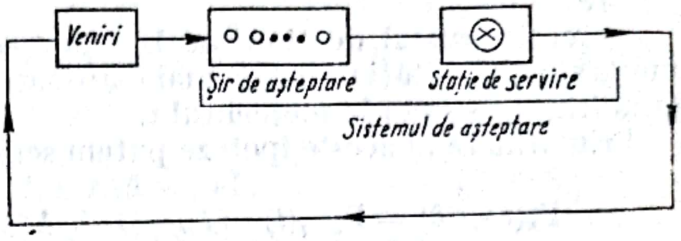


Fig. 7.1.

De exemplu, într-un atelier dintr-o întreprindere funcționează N mașini omogene, cu aceleași caracteristici tehnologice și care lucrează independent unele de altele. Ele se pot defecta. Numărul mediu de defecțiuni în unitatea de timp fiind, pentru o mașină, egal cu λ. Pentru remedierea defecțiunilor presupunem că atelierul dispune de un mecanic calificat — ceea ce înseamnă existența unei singure stații de reparare — și că durata unei reparații este o variabilă aleatoare ce urmează o repartiție exponențială negativă de parametru μ.

În acest caz factorul de serviciu $\rho = \frac{\lambda}{\mu}$ mai poartă și numele de factor de întreținere.

Schematic modul de funcționare al unui astfel de sistem poate fi reprezentat după cum urmează :

Admitem că mașinile se repară pe măsură ce se defectează și deci nici una dintre ele nu are prioritate, ceea ce înseamnă că se respectă disciplina sistemului : „primul venit, primul servit".

Pentru a pune în evidență ecuațiile ce caracterizează starea sistemului, vom căuta să exprimăm probabilitatea ca la momentul $t+\delta$ să existe n unități în sistem și pe care o vom nota $\mathbf{P}_n(t+\delta)$.

Cum $1-\mathbf{A}(\delta)$ este probabibilitatea ca o mașină să nu se defecteze într-un interval de timp de lungime δ și cum defecțiunile apar independent unele de altele, rezultă că probabilitatea ca nici o mașină din cele $N-n$ care funcționează să nu se defecteze este :

$$[1 - \mathbf{A}(\delta)]^{N-n}$$

De aici rezultă că probabilitatea să se defecteze cel puțin una din cele $N - n$ (care sînt în stare de funcționare) este

$$1 - [1 - \mathbf{A}(\delta)]^{N-n} = (N - n)\,\lambda\delta + \mathbf{O}(\delta).$$

Probabilitatea ca să fie terminată o reparație într-un interval de lungime δ în ipotezele formulate este

$$\mathbf{B}(\delta) = \mu\delta + \mathbf{O}(\delta).$$

Scanned with OKEN Scanner

Acum putem exprima evenimentul ca la momentul $t + \delta$ să avem n unităţi în sistem, ca reuniunea următoarelor evenimente incompatibile două cîte două :

— la momentul t există n unităţi în sistem şi în intervalul de timp de la t la $t + \delta$ să nu înregistrăm nici o venire şi nici o plecare.

— la momentul t să avem $n - 1$ unităţi în sistem şi în intervalul de timp de la t la $t + \delta$ să înregistrăm o singură venire şi nici o plecare ;

— la momentul t să avem $n + 1$ unităţi în sistem şi în intervalul de timp de la t la $t + \delta$ să nu înregistrăm nici o venire şi doar o singură plecare.

— în intervalul de timp de la t la $t + \delta$ să înregistrăm o variaţie a numărului de unităţi în sistem mai mare sau egală cu 2, oricare ar fi numărul de unităţi în sistem la momentul t.

Prin urmare în aceste ipoteze putem scrie

$$\mathbf{P}_n(t+\delta) = \mathbf{P}_{n-1}(t)\ [1 - (1 - \mathbf{A}(\delta))^{N-n+1}]\ [1 - \mathbf{B}(\delta)] +$$

$$+ \mathbf{P}_n(t)\ [1 - \mathbf{A}(\delta)]^{N-n}\ [1 - \mathbf{B}(\delta)] + \mathbf{P}_{n+1}(t)\ [1 - \mathbf{A}(\delta)]^{N-n-1}\mathbf{B}(\delta) + \mathbf{O}(\delta).$$

Înlocuind pe $\mathbf{A}(\delta)$ şi $\mathbf{B}(\delta)$, efectuînd calculele şi împărţind cu δ obţinem :

$$\frac{\mathbf{P}_n(t+\delta) - \mathbf{P}_n(t)}{\delta} = (N - n + 1)\,\lambda\,\mathbf{P}_{n-1}(t) - [(N - n)\,\lambda + \mu]\,\mathbf{P}_n\,(t) +$$

$$+ \mu\,\mathbf{P}_{n+1}(t) + \frac{\mathbf{O}(\delta)}{\delta}$$

Făcînd pe $\delta \to 0$, membrul al doilea are limită, deci şi membrul întîi are limită şi avem :

$$\frac{d\,\mathbf{P}_n(t)}{dt} = (N - n + 1)\,\lambda\,\mathbf{P}_{n+1}(t) - [(N - n)\,\lambda + \mu]\,\mathbf{P}_n\,(t) + \mu\mathbf{P}_{n+1}(t)$$

$$n = 1, 2, \ldots, N - 1.$$

Dacă $n = 0$, un raţionament analog ne conduce la

$$\mathbf{P}_0(t+\delta) = \mathbf{P}_0(t)\,[1 - \mathbf{A}(\delta)]^N + \mathbf{P}_1(t)\,[1 - \mathbf{A}(\delta)]^{N-1}\mathbf{B}(\delta) + \mathbf{O}(\delta)$$

de unde se obţine ecuaţia :

$$\frac{d\,\mathbf{P}_0(t)}{dt} = -\,N\,\lambda\,\mathbf{P}_0(t) + \mu\mathbf{P}_1(t)$$

În fine dacă $n = N$ avem :

$$\mathbf{P}_N(t+\delta) = \mathbf{P}_{N-1}(t)\,\mathbf{A}(\delta)\,[1 - \mathbf{B}(\delta)] + \mathbf{P}_N(t)\,[1 - \mathbf{B}(\delta)] + \mathbf{O}(\delta).$$

de unde se obţine :

$$\frac{d\,\mathbf{P}_N(t)}{dt} = \lambda\,\mathbf{P}_{N-1}(t) - \mu\,\mathbf{P}_N(t).$$

Scanned with OKEN Scanner

Vom presupune că ne situăm în cazul echilibrului statistic :

$$\lim_{t\to\infty} \frac{d\,P_n(t)}{dt} = 0 \quad n = 0, 1, \ldots, N;$$

și atunci ecuațiile probabilităților modelului sînt date de sistemul de ecuații algebrice liniare :

$$\begin{cases} -N\lambda p_0 + \mu p_1 = 0 \\ (N-n+1)\lambda p_{n-1} - [(N-n)\lambda+\mu]\ p_n + \mu p_{n+1} = 0 \quad n=1, 2, \ldots, N-1 \\ \lambda p_{N-1} - \mu p_N = 0 \end{cases}$$

Cea de a doua relație mai poate fi scrisă astfel :

$$(N - n + 1)\lambda p_{n-1} - \mu p_n = (N - n)\lambda p_n - \mu p_{n+1},$$
$$n = 1, 2, \ldots, N - 1$$

Notînd

$$z_n = (N - n + 1)\lambda p_{n+1} - \mu p_n,$$

avem relația :

$$z_n = z_{n+1}, \quad n = 1, 2, \ldots, N - 1.$$

Cum $z_1 = 0$ rezultă $z_n = 0 \quad n = 1, 2, \ldots, N.$

și deci

$$p_n = (N - n + 1)\ \rho\, p_{n-1}, \quad n = 1, 2, \ldots, N$$

de unde urmează

$$p_n = \frac{N!}{(N-n)!}\rho^n p_0.$$

Probabilitatea p_0 se obține imediat din relația

$$\sum_{n=0}^{N} \frac{N!}{(N-n)!}\rho^n p_0 = 1.$$

Deci

$$p_0 = \frac{1}{1 + \sum_{n=1}^{N} \frac{N!}{(N-n)!}\rho^n}$$

Putem obține acum imediat caracteristicile numerice ale modelului. Din repartiția variabilei aleatoare Z — numărul de unități în sistem :

$$Z : \begin{pmatrix} 0 & 1 & 2 & \ldots & n & \ldots & N \\ p_0 & p_1 & p_2 & \ldots & p_n & \ldots & p_N \end{pmatrix}$$

Scanned with OKEN Scanner

obținem :

$$\bar{Z} = M(Z) = \sum_{n=0}^{N} n\, p_n = \sum_{n=0}^{N} n \frac{N!}{(N-n)!} \rho^n p_0.$$

Înlocuind pe p_0 și efectuînd unele calcule algebrice se obține expresia :

$$\bar{Z} = N - \frac{1}{\rho}(1 - p_0).$$

Pe baza acestei valori medii putem obține numărul mediu de unități în șirul de așteptare :

$$\bar{Q} = M(Q) = \sum_{n=1}^{N} (n-1)\, p_n = \bar{Z} - (1 - p_0)$$

Înlocuind pe $\bar{Z}$ cu valoarea mai sus determinată se obține :

$$\bar{Q} = N - \frac{\rho + 1}{\rho}(1 - p_0).$$

Media de inactivitate a stației este

$$\bar{\beta} = \sum_{n=0}^{1} (1 - n)\, p_n = p_0.$$

O altă caracteristică importantă din punct de vedere economic îl reprezintă numărul mediu de unități ce nu se găsesc în sistem (în cazul concret al exemplului nostru această caracteristică reprezintă numărul mediu de mașini în stare de funcționare). Această valoare medie este dată de :

$$M(N - Z) = \sum_{n=0}^{N} (N - n)\, p_n = N - \sum_{n=0}^{N} n\, p_n = N - \bar{Z} = \frac{1}{\rho}(1 - p_0)$$

Probabilitatea ca o unitate să aștepte în șir, adică probabilitatea ca stația să fie ocupată este

$$p(>0) = \sum_{n=1}^{N} p_n = 1 - p_0$$

Pentru a calcula timpul mediu de așteptare în șir vom face mai întîi observația că în cazul regimului permanent de funcționare, media venirilor nu mai este λ ci $\lambda M(N - Z)$ adică $\lambda (N - \bar{Z})$. De aici urmează că

$$\bar{Q} = \lambda(N - \bar{Z})W$$

Scanned with OKEN Scanner

și deci

$$W = \frac{1}{\lambda(N-\bar{Z})}\bar{Q} = \frac{1}{\lambda(N-\bar{Z})}\left[N - \frac{1+\rho}{\rho}(1-p_0)\right]$$

relație care mai poate fi pusă și sub forma

$$W = \frac{1}{\mu}\left[\frac{N}{1-p_0} - \frac{1+\rho}{\rho}\right]$$

Timpul mediu de așteptare în sistem va fi :

$$W^* = W + \frac{1}{\mu} = \frac{1}{\mu}\left[\frac{N}{1-p_0} - \frac{1+\rho}{\rho} + 1\right] = \frac{\bar{Z}}{\mu\rho(N-\bar{Z})}$$

Exemplul 7.4. Într-un atelier al unei întreprinderi funcționează 6 mașini automate care se defectează în mod aleator și un muncitor cu calificare superioară le repune în stare de funcționare. Din măsurători statistice s-a constatat că numărul de defecțiuni este o variabilă aleatoare repartizată Poisson cu parametrul $\lambda = 2$ defecțiuni pe oră iar durata unei reparații este o variabilă aleatoare repartizată exponențial negativ cu media 15 minute.

Să se calculeze :

— probabilitatea ca la un moment dat toate mașinile să funcționeze ;
— probabilitatea ca să fie defecte la un moment dat cel mult 3 mașini ;
— probabilitatea ca la un moment dat toate mașinile să fie defecte ;
— numărul mediu de mașini defecte la un moment dat ;
— numărul mediu de mașini ce așteaptă pentru a fi repuse în stare de funcționare la un moment dat ;
— timpul mediu de așteptare în șir ;
— timpul mediu de așteptare în sistem.

Soluție : Folosim un model de așteptare cu o singură stație cu veniri dintr-o populație finită.

Alegem unitatea de timp ora și atunci $\mu = 4$. Deci $\rho \simeq \frac{2}{4} = \frac{1}{2}$.

Atunci probabilitatea ca la un moment dat toate mașinile să funcționeze este :

$$p_0 = \frac{1}{1 + \sum_{n=1}^{N} \frac{N!}{(N-n)!}\rho^n} \simeq 0{,}012.$$

Probabilitatea ca toate mașinile să fie defecte este $p_6 = 0{,}135$. deci în aproximativ 13% din cazuri toate mașinile sînt defecte.

Probabilitatea ca la un moment dat să fie defecte cel mult 3 mașini este

$$p_0 + p_1 + p_2 + p_3 = 0{,}318.$$

Numărul mediu de mașini defecte la un moment dat

$$\bar{Z} = N - \frac{1}{\rho}(1-p_0) = 4{,}024$$

Scanned with OKEN Scanner

adică în medie patru mașini.

Numărul mediu de mașini ce așteaptă să fie reparate

$$\bar{Q} = N - \frac{\rho + 1}{\rho}(1 - p_0) = 3{,}036$$

adică în medie 3 mașini așteaptă pentru a fi reparate

Timpul mediu de așteptare în șir :

$$W = \frac{\bar{Q}}{\lambda(N - \bar{Z})} = 0{,}768 \text{ ore}$$

sau 46 minute.

Timpul mediu de așteptare în sistem

$$W^* = 46 + 15 = 61 \text{ minute.}$$

7.4. Modele de așteptare cu mai multe stații de servire

Modelele pe care le-am studiat pînă acum aveau drept caracteristică comună stația unică de servire. În foarte multe situații practice, modelele cu o singură stație devin nesatisfăcătoare, așa că sîntem nevoiți să considerăm și să studiem fenomenul de aglomerare în care se utilizează mai multe stații de servire. În cazul modelelor de așteptare cu mai multe stații apar probleme noi, atît în ceea ce privește obținerea sistemului de ecuații din care se obțin probabilitățile stărilor sistemului cît și în ceea ce privește latura concret economică. De exemplu se poate pune problema determinării numărului minim de stații astfel încît serviciul să se desfășoare în condiții optime pentru solicitanți și cu cheltuieli cît mai mici pentru întreținerea în stare de funcționare a sistemului.

În cele ce urmează vom aborda doar două modele cu mai multe stații de servire și anume un model cu veniri dintr-o populație infinită (model deschis) și un model cu veniri dintr-o populație finită (model închis).

7.4.1. Model de așteptare cu mai multe stații de servire și cu veniri dintr-o populație infinită

Ipotezele de lucru pentru acest model sînt următoarele : modelul are s stații dispuse în paralel ce funcționează independent unele de altele, independent de veniri și că durata serviciului pentru fiecare dintre cele s stații este o variabilă aleatoare repartizată exponențial negativ de parametru μ.

Venirile sînt întîmplătoare, independente între ele și independente de plecări și sînt poissoniene de parametru λ.

Scanned with OKEN Scanner

Cînd o staţie este liberă, prima unitate din şir intră în staţia respectivă pentru a primi serviciul solicitat. Prin urmare disciplina în sistem este cea naturală „primul venit" este „primul servit", fără a exista o preferinţă pentru o anumită staţie şi fără a exista unităţi preferate la servire, în dauna altor unităţi.

Schematic un astfel de model se poate reprezenta ca în figura de mai jos :

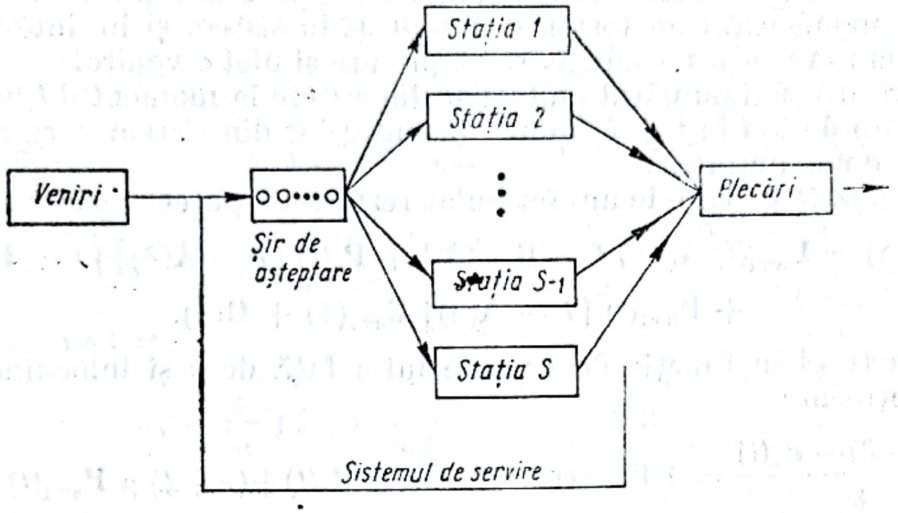


Fig. 7.2.

Vom căuta să scriem sistemul de ecuaţii ale probabilităţilor stărilor sistemului pentru modelul de aşteptare cu s staţii în ipotezele pe care le-am formulat mai înainte. Pentru aceasta vom exprima mai întîi probabilităţile ca o unitate să intre în sistem, respectiv să părăsească sistemul.

Am văzut deja în cazul modelelor cu o singură staţie că probabilitatea $\mathbf{A}(\delta)$ ca într-un interval de timp de lungime $\delta > 0$ să avem o venire are expresia :

$$\mathbf{A}(\delta) = \lambda\delta + \mathbf{0}(\delta),$$

iar probabilitatea ca o unitate să părăsească o staţie are expresia :

$$\mathbf{B}(\delta) = \mu\delta + \mathbf{0}(\delta).$$

Dacă presupunem că în sistem se găsesc n unităţi atunci în cazul unui model cu s staţii vom întîlni două situaţii şi anume :

$a)$ $0 \leqslant n < s$

$b)$ $n \geqslant s$

Dacă $0 \leqslant n < s$, fiecare din cele n unităţi poate părăsi sistemul într-un interval de timp de lungime δ şi această probabilitate are expresia :

$$\mathbf{B}_n(\delta) = 1 - [1 - \mathbf{B}(\delta)]^n = n\,\mathbf{B}(\delta) + \mathbf{0}(\delta).$$

În cazul cînd toate staţiile sînt ocupate, adică $n \geqslant s$, probabilitatea ca o unitate să părăsească sistemul va fi :

$$\mathbf{B}_n(\delta) = s\,\mathbf{B}(\delta) + \mathbf{0}(\delta), \quad n = s,\ s+1, \ldots$$

Vom exprima probabilitatea $\mathbf{P}_n(t+\delta)$ atît în cazul $0 \leqslant n < s$ cît și în cazul $n \geqslant s$.

Pentru aceasta vom descompune evenimentul a cărui probabilitate este $\mathbf{P}_n(t+\delta)$ în următoarele evenimente incompatibile :

— la momentul t au fost n unități în sistem și în intervalul de timp de la t la $t+\delta$ nu a avut loc nici o venire și nici o plecare ;

— la momentul t au fost $n-1$ unități în sistem și în intervalul de timp de la t la $t+\delta$ s-a produs o venire și nu a avut loc nici o plecare.

— la momentul t au fost $n+1$ unități în sistem și în intervalul de timp de la t la $t+\delta$ s-a înregistrat o plecare și nici o venire.

— oricare ar fi numărul unităților din sistem la momentul t în intervalul de timp de la t la $t+\delta$, numărul unităților din sistem variază cu mai mult de două unități.

În ipotezele pe care le-am formulat rezultă că putem scrie :

$$\mathbf{P}_n(t+\delta) = \mathbf{P}_{n-1}(t)\,\mathbf{A}(\delta)\,[1-\mathbf{B}_{n-1}(\delta)] + \mathbf{P}_n(t)\,[1-\mathbf{A}(\delta)]\,[1-\mathbf{B}_n(\delta)] + \\ + \mathbf{P}_{n+1}(t)\,[1-\mathbf{A}(\delta)]\,\mathbf{B}_{n+1}(\delta) + \mathbf{O}(\delta).$$

Particularizînd în funcție de valorile lui n față de s și înlocuind $\mathbf{A}(\delta)$ și $\mathbf{B}_n(\delta)$ obținem :

$$\frac{\mathbf{P}_n(t+\delta)-\mathbf{P}_n(t)}{\delta} = \lambda\,\mathbf{P}_{n-1}(t) - (\lambda + n\,\mu)\,\mathbf{P}_n(t) + (n+1)\,\mu\,\mathbf{P}_{n+1}(t) + \frac{\mathbf{O}(\delta)}{\delta}$$

$$n = 1,\ 2,\ldots,\ s-1$$

$$\frac{\mathbf{P}_n(t+\delta)-\mathbf{P}_n(t)}{\delta} = \lambda\,\mathbf{P}_{n-1}(t) - (\lambda + s\,\mu)\,\mathbf{P}_n(t) + s\,\mu\,\mathbf{P}_{n+1}(t) + \frac{\mathbf{O}(\delta)}{\delta}$$

$$n = s,\ s+1,\ldots$$

În cazul $n = 0$ putem scrie

$$\mathbf{P}_0(t+\delta) = \mathbf{P}_0(t)\ [1-\mathbf{A}(\delta)] + \mathbf{P}_1(t)\ [1-\mathbf{A}(\delta)]\ \mathbf{B}_1(\delta) + \mathbf{O}(\delta)$$

de unde

$$\frac{\mathbf{P}_0(t+\delta)-\mathbf{P}_0(t)}{\delta} = -\lambda\,\mathbf{P}_0(t) + \mu\,\mathbf{P}_1(t) + \frac{\mathbf{O}(\delta)}{\delta}$$

Făcînd pe $\delta \to 0$ se obține sistemul de ecuații diferențiale :

$$\frac{\mathrm{d}\,\mathbf{P}_0(t)}{\mathrm{d}t} = -\lambda\,\mathbf{P}_0(t) + \mu\,\mathbf{P}_1(t)$$

$$\frac{\mathrm{d}\,\mathbf{P}_n(t)}{\mathrm{d}t} = \lambda\,\mathbf{P}_{n-1}(t) - (\lambda + n\,\mu)\ \mathbf{P}_n(t) + (n+1)\,\mu\,\mathbf{P}_{n+1}(t)$$

$$n = 1,\ 2,\ldots,\ s-1$$

$$\frac{\mathrm{d}\,\mathbf{P}_n(t)}{\mathrm{d}t} = \lambda\,\mathbf{P}_{n-1}(t) - (\lambda + s\,\mu)\,\mathbf{P}_n(t) + s\,\mu\,\mathbf{P}_{n+1}(t)$$

$$n = s,\ s+1,\ldots$$

Scanned with OKEN Scanner

Vom presupune mai departe că sîntem în cazul regimului staţionar de funcţionare, adică

$$\lim_{t\to\infty} \frac{d\,P_n(t)}{dt} = 0.$$

În acest caz rezultă că probabilităţile stărilor verifică sistemul de ecuaţii algebrice :

$$\begin{cases} -\lambda p_0 + \mu p_1 = 0 \\ \lambda p_{n-1} - (\lambda + n\mu) p_n + (n+1)\mu p_{n+1} = 0 \quad n = 1, 2, \ldots, s-1 \\ \lambda p_{n-1} - (\lambda + s\mu) p_n + s\mu p_{n+1} = 0 \quad n = s, s+1, \ldots \end{cases}$$

Din ecuaţiile $\lambda p_{n-1} - (\lambda + n\mu) p_n + (n+1)\mu p_{n+1} = 0$

$$n = 1, 2, \ldots, s-1.$$

se obţine imediat

$$p_n = \frac{\rho}{n} p_{n-1} = \frac{\rho^n}{n!} p_0 \quad n = 1, 2, \ldots, s.$$

Dacă în ultimul grup de ecuaţii facem $n = s$ obţinem

$$p_{s+1} = \frac{1}{s\mu} [(\lambda + s\mu)\ p_s - \lambda p_{s-1}] = \frac{\rho^{s+1}}{s \cdot s!} p_0.$$

Făcînd apoi pe $n = s+1,\ s+2, \ldots$ se obţine

$$p_{s+r} = \frac{\rho^{s+r}}{s^r \cdot s!} p_0$$

sau

$$p_n = \frac{\rho^n}{s!\, s^{n-s}} p_0 \quad n = s, s+1, \ldots$$

Deci

$$p_n = \begin{cases} \dfrac{\rho^n}{n!} p_0, & \text{dacă } 0 \leqslant n < s \\ \dfrac{\rho^n}{s!\, s^{n-s}} p_0, & \text{dacă } n \geqslant s \end{cases}$$

Probabilitatea p_0 se determină punînd condiţia :

$$\sum_{n=0}^{\infty} p_n = 1.$$

Scanned with OKEN Scanner

Deci

$$1 = p_0\left(\sum_{n=0}^{s-1}\frac{\rho^n}{n!}+\frac{\rho^s}{s!}\frac{1}{1-\frac{\rho}{s}}\right)$$

sau

$$p_0 = \frac{1}{\sum_{n=0}^{s-1}\frac{\rho^n}{n!}+\frac{\rho^s}{s!\,(1-\rho^*)}}$$

unde am pus $\rho = \frac{\lambda}{\mu}$ și $\rho^* = \frac{\rho}{s}$ care este numit și factorul de serviciu al sistemului.

Avînd probabilităţile stărilor putem obţine o serie de caracteristici numerice utile în analize economice. Mai întîi numărul mediu de unităţi în şirul de aşteptare :

$$\bar{\mathbf{Q}} = \mathbf{M}(\mathbf{Q}) = \sum_{n=s+1}^{\infty}(n-s)\,p_n = p_0\frac{\rho^s}{s!}\sum_{n=s+1}^{\infty}(n-s)\frac{\rho^{n-s}}{s^{n-s}} =$$

$$= p_0\frac{\rho^s}{s!}\frac{\rho^*}{(1-\rho^*)^2}$$

Dacă notăm cu **L** numărul de staţii neocupate (libere) avem :

$$\bar{\mathbf{L}} = \mathbf{M}(\mathbf{L}) = \sum_{n=0}^{s-1}(s-n)\,p_n = \sum_{n=0}^{s-1}(s-n)\frac{\rho^n}{n!}p_0 =$$

$$= p_0(s-\rho)\left[\sum_{n=0}^{s-1}\frac{\rho^n}{n!}+\frac{\rho^s}{s!}\frac{1}{1-\rho^*}\right]$$

Ţinînd cont şi de expresia probabilităţii p_0 se obţine în final :

$$\bar{\mathbf{L}} = s - \rho.$$

Pentru a obţine numărul mediu de unităţi în sistem $\bar{\mathbf{Z}}$ vom utiliza relaţia imediată :

$$\mathbf{Z} = \mathbf{Q} + s - \mathbf{L}$$

(numărul unităţilor din sistem este suma dintre numărul unităţilor în şir şi numărul staţiilor ocupate).

De aici urmează că

$$\bar{\mathbf{Z}} = \mathbf{M}(\mathbf{Z}) = \bar{\mathbf{Q}} + s - \bar{\mathbf{L}} = p_0\frac{\rho^s}{s!}\frac{\rho^*}{(1-\rho^*)^2} + \rho.$$

Scanned with OKEN Scanner

Timpul mediu de aşteptare în şir este

$$W = \frac{\overline{Q}}{\lambda} = \frac{p_0 \frac{\rho^s}{s!} \cdot \frac{\rho^*}{(1-\rho^*)^2}}{\lambda} = p_0 \frac{\rho^s}{s \cdot s!} \cdot \frac{1}{(1-\rho^*)^2 \mu}$$

Exemplul 7.5. Într-o intreprindere se pune problema stabilirii numărului optim de magazioneri care trebuie să distribuie muncitorilor din secţii materii prime şi materiale. Pentru aceasta s-a studiat statistica sosirilor muncitorilor şi a timpului utilizat de magazioner pentru furnizarea materialelor. În acest scop s-au efectuat 120 de măsurători asupra numărului de solicitanţi pe un interval de timp fixat de 10 minute şi s-au înregistrat datele din tabelul 7.1.

Tabelul 7.1

Nr. crt.	Numărul de sosiri în intervalul de 10 minute	Frecvenţa absolută observată
1	5	3
2	6	4
3	7	4
4	8	5
5	9	6
6	10	6
7	11	7
8	12	6
9	13	8
10	14	12
11	15	10
12	16	12
13	17	9
14	18	8
15	19	5
16	20	5
17	21	4
18	22	3
19	23	1
20	24	1
21	25	1

Măsurîndu-se duratele de servire s-a constatat că durata medie unei serviri este de 2 minute.

De asemenea s-a verificat cu ajutorul criteriului χ^2 că numărul de veniri pe unitatea de timp (în cazul nostru 10 minute) este o variabilă Poisson, iar durata serviciului este o variabilă aleatoare repartizată exponenţial negativ.

Numărul de magazioneri se stabileşte astfel încît pierderile cauzate de inactivitatea muncitorilor ce aşteaptă să fie serviţi şi de inactivitatea magazionerilor ce nu au ce servi să fie minime.

S-a evaluat că retribuţia medie orară a unui muncitor este de 10 unităţi monetare iar a unui magazioner este de 8 unităţi monetare.

Scanned with OKEN Scanner

Soluție: Din datele problemei rezultă că numărul mediu de sosiri pe intervalul de 10 minute este

$$\lambda = \frac{1697}{120} \simeq 14$$

Cum durata medie a unei serviri este de 2 minute rezultă că pe intervalul de *10* minute sînt satisfăcute în medie

$$\mu = \frac{10}{2} = 5 \text{ cereri de materiale}$$

Urmează că

$$\rho = \frac{\lambda}{\mu} = \frac{14}{5} > 1$$

și deci un singur magazioner nu este suficient, acesta conducînd la un fenomen de aglomerare continuă.

Vom utiliza un model de așteptare cu mai multe stații cu veniri dintr-o populație infinită și vom evalua cheltuielile pe ziua de lucru de *8* ore pentru $s = 2, 3, 4, \ldots$

Pentru $s = 2$ rezultă $\rho^* = \frac{\rho}{2} = \frac{14}{10} > 1$ deci și în acest caz se menține fenomenul de aglomerare.

Pentru $s = 3$, $\rho^* = \frac{\rho}{3} = \frac{14}{15} < 1.$

$$p_0 = \frac{1}{\sum_{n=0}^{s-1} \frac{\rho^n}{n!} + \frac{\rho^s}{s!(1-\rho^*)}} \text{ devine } p_0 = \frac{1}{1 + \frac{\rho}{1!} + \frac{\rho^2}{2!} + \frac{\rho^*}{3!\left(1-\frac{14}{15}\right)}} = 0{,}015$$

Calculăm timpul mediu de așteptare în șir :

$$W = p_0 \frac{\rho^s}{s \cdot s!} \frac{1}{(1-\rho^*)^2 \mu} = 0{,}823 \text{ unități de timp}$$

deci $W = 8{,}23$ minute.

Timpul mediu pierdut de toți solicitanții în *8* ore :

$$\lambda \times 8{,}23 \times 48 = 14 \cdot 8{,}23 \cdot 48 = 5530{,}56 \text{ minute}$$

$$\simeq 92{,}18 \text{ ore}$$

$C_3 = 92{,}18 \times 10 = 921{,}8$ unități monetare.

Scanned with OKEN Scanner

Cheltuielile de nefuncţionare a sistemului de servire. Numărul mediu de solicitanţi într-o zi de lucru (*8* ore) este

$$\lambda \times 6 \times 8 = 14 \cdot 6 \cdot 8 = 672.$$

Pentru cei 672 de solicitanţi este nevoie de $\frac{672}{\mu}$ unităţi de timp a *10* minute adică *134,4* unităţi de *10* minute sau *1344* minute.

Timpul total disponibil al celor 3 magazioneri în *8* ore este: $3 \times 60 \times 8 = 1440$ minute.

Durata medie de inactivitate pentru cei 3 magazioneri este $1440 - 1344 = 96$ minute adică *1,6* ore care este evaluat la $1,6 \times 8 = 12,8$ unităţi monetare.

Urmează că costul total de inactivitate se evaluează pentru 3 staţii la

$$F(3) = C_3 + C_3' = 921,8 + 12,8 = 934,6$$

unităţi monetare.

Pentru $s = 4$

$$p_0 = 0,05$$

$$W = 0,032 \text{ unităţi de timp}$$

deci

$$W = 0,32 \text{ minute.}$$

Timpul mediu pierdut de solicitanţi în *8* ore este $14 \cdot 0,32 \times 48 = 215,04$ minute sau *3,584* ore. Urmează că cheltuielile datorate inactivităţii muncitorilor sînt

$$C_4 = 3,584 \cdot 10 = 35,84.$$

Cheltuielile de nefuncţionare a sistemului de servire

$$4 \cdot 60 \cdot 8 = 1920 \text{ minute}$$

$$1920 - 1344 = 576 \text{ minute sau } 9,6 \text{ ore}$$

şi de aici urmează

$$C_4' = 9,6 \times 8 = 76,8 \text{ unităţi monetare}$$

$$F(4) = 35,84 + 76,8 = 112,64$$

Pentru $s = 5$

$$P_0 = 0,005$$

$$W = 0,0015 \text{ unităţi de timp}$$

deci

$$W = 0,015 \text{ minute}$$

Timpul mediu pierdut de solicitanţi în *8* ore

$$14 \cdot 0,015 \cdot 48 = 10,08 \text{ minute, adică } 0,168 \text{ ore}$$

$$C_5 = 0,168 \times 10 = 1,68 \text{ unităţi monetare.}$$

Scanned with OKEN Scanner

Cheltuielile de nefuncționare a sistemului de servire:

$5 \cdot 60 \cdot 8 = 2400$

$2400 - 1344 = 1056$ minute sau $17{,}60$ ore.

$C'_5 = 17{,}60 \times 8 = 140{,}80$

$F(5) = 1{,}68 + 140{,}80 = 142{,}48$ unități monetare.

Pentru $s > 5$ cheltuielile cresc și deci numărul optim de magazioneri necesar este $s = 4$.

Se poate calcula imediat probabilitatea ca o unitate să aștepte în sistem. Cum o unitate intrată în sistem va avea de așteptat pentru a fi servită numai dacă $Z \geqslant s$ rezultă că:

$$P(Z \geqslant s) = \sum_{n=s}^{\infty} p_n = p_0 \frac{\rho^s}{s!} \frac{1}{1-\rho^*}$$

7.4.2. Model de așteptare cu mai multe stații de servire și unitățile provenind dintr-o populație finită

Datorită multiplelor posibilități de aplicare — îndeosebi în industrie — acest model este mult studiat. Pentru o mai bună înțelegere a modului cum se pune problema vom apela la o anumită situație concretă.

Presupunem că este vorba de o secție sau un atelier al unei anumite uzine în care funcționează N mașini omogene (eventual identice) și că în acest atelier sînt încadrați s mecanici care intervin atunci cînd o mașină s-a defectat, sau nu mai lucrează la nivelul parametrilor specificați de procesul tehnologic, necesitînd deci o reparație sau un reglaj.

Schematic, un astfel de model se poate reprezenta astfel:

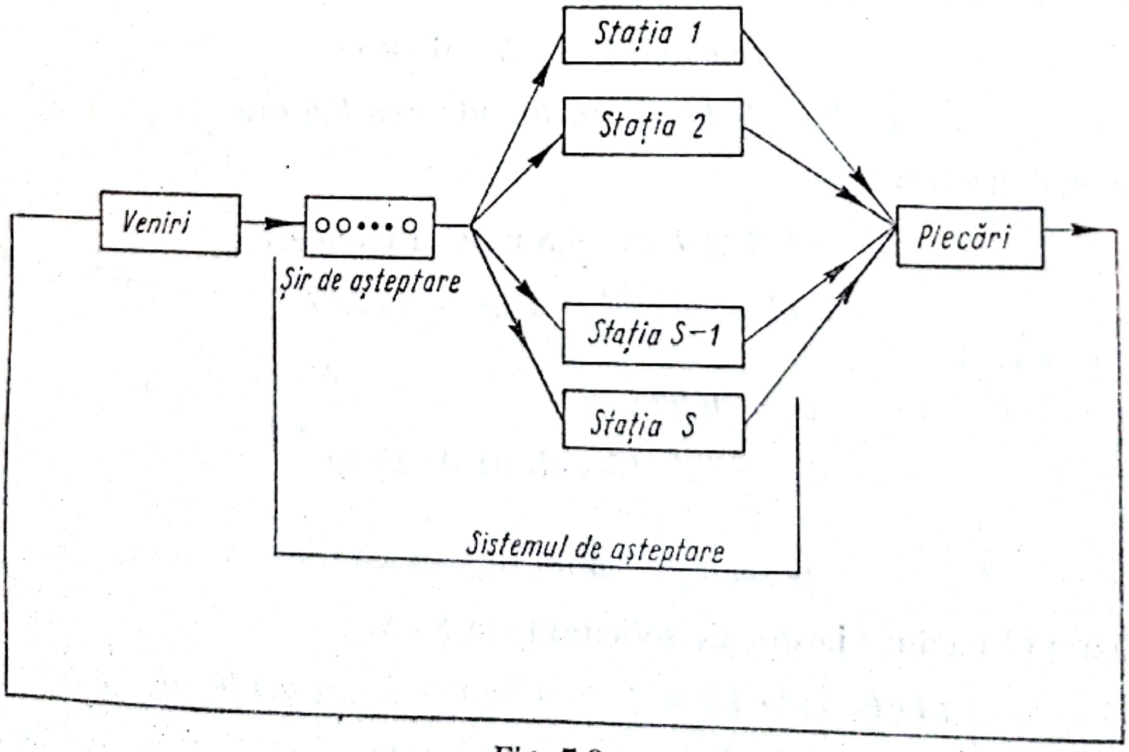


Fig. 7.3.

Scanned with OKEN Scanner

Ipotezele care stau la baza modelului sînt următoarele :

a) durata de servire pentru fiecare staţie este o variabilă aleatoare repartizată exponenţial negativ de parametru μ.

b) venirile sînt întîmplătoare, independente între ele şi independente de servire şi că numărul de veniri pe unitatea de timp este o variabilă aleatoare repartizată Poisson.

c) admitem că disciplina sistemului este cea obişnuită adică „primul venit" este „primul servit".

Admiţînd aceste ipoteze putem scrie imediat probabilitatea de trecere de la o stare n la o stare $n + k$ după cum urmează :

— Dacă la momentul t sistemul s-a aflat în starea n el poate rămîne în aceeaşi stare la momentul $t + \delta$ cu probabilitatea :

$$[1 - (N - n)\,\lambda\,\delta + \mathbf{0}(\delta)]\;[1 - n\,\mu\,\delta + \mathbf{0}(\delta)]$$

dacă $0 < n < s$ şi cu probabilitatea

$$[1 - (N - n)\,\lambda\,\delta + \mathbf{0}(\delta)]\;[1 - s\,\mu\,\delta + \mathbf{0}(\delta)]$$

dacă $s \leqslant n < N$.

— Dacă la momentul t sistemul s-a aflat în starea $n + 1$ el poate trece în starea n la momentul $t + \delta$ cu probabilitatea :

$$[1 - (N - n - 1)\;\lambda\,\delta + \mathbf{0}(\delta)]\;[(n + 1)\;\mu\,\delta + \mathbf{0}(\delta)]$$

dacă $0 < n < s$ şi cu probabilitatea

$$[1 - (N - n - 1)\;\lambda\,\delta + \mathbf{0}(\delta)]\;[s\,\mu\,\delta + \mathbf{0}(\delta)]$$

dacă $s \leqslant n < N$.

— Dacă la momentul t sistemul s-a aflat în starea $n - 1$ el poate trece în starea n la momentul $t + \delta$ cu probabilitatea :

$$[(N - n + 1)\,\lambda\,\delta + \mathbf{0}(\delta)]\;[1 - (n - 1)\,\mu\,\delta + \mathbf{0}(\delta)]$$

dacă $0 < s < n$ şi cu probabilitatea

$$[(N - n + 1)\,\lambda\,\delta + \mathbf{0}(\delta)]\;[1 - \tilde{s}\mu\,\delta + \mathbf{0}(\delta)]$$

dacă $s \leqslant n < N$[1]

— Din orice altă stare j diferită de cele amintite sistemul poate trece în starea n cu probabilitatea $\mathbf{0}(\delta)$. Pe baza acestor probabilităţi de trecere putem exprima acum probabilitatea $\mathbf{P}_n(+\delta)$ probabilitatea ca la momentul $t + \delta$ sistemul să se afle în starea n — (sînt n unităţi în sistem).

$$\mathbf{P}_n(t+\delta) = (N - n + 1)\,\lambda\,\delta\,\mathbf{P}_{n-1}(t) + [1 - \{(N - n)\,\lambda + n\,\mu\}\,\delta]\,\mathbf{P}_n(t) + \\ + (n + 1)\,\mu\,\delta\,\mathbf{P}_{n+1}(t) + \mathbf{0}(\delta).$$

dacă

$$1 \leqslant n \leqslant s - 1$$

şi

$$\mathbf{P}_n(t + \delta) = (N - n + 1)\,\lambda\,\delta\,\mathbf{P}_{n-1}(t) + \\ + [1 - \{(N - n)\,\lambda + s\,\mu\}\,\delta]\,\mathbf{P}_n(t) + s\,\mu\,\delta\,\mathbf{P}_{n+1}(t) + \mathbf{0}(\delta).\quad s \leqslant n < N$$

[1] Am pus $\tilde{s}$ pentru a pune în evidenţă că este vorba de $s - 1$ dacă $n = s$ sau s dacă $n > s$.

Scanned with OKEN Scanner

Dacă $n = 0$, $P_0(t + \delta)$ se exprimă după cum urmează

$$P_0(t + \delta) = (1 - N \lambda \delta) P_0(t) + \mu \delta P_1(t) + O(\delta).$$

iar dacă $n = N$ (adică toate unităţile populaţiei considerate se găsesc în sistem) putem scrie :

$$P_N(t + \delta) = P_{N-1}(t) \ [\lambda \delta + O(\delta)] \ [1 - s \mu \delta + O(\delta)] + + P_N(t) \ [1 - s \mu \delta + O(\delta)] + O(\delta).$$

Trecînd acum $P_n(t)$ în primul membru, împărţind apoi cu δ în ambii membri şi făcînd $\delta \to 0$ se obţine următorul sistem de ecuaţii diferenţiale din care se determină probabilităţile stărilor :

$$\frac{dP_0(t)}{dt} = - N \lambda P_0(t) + \mu P_1(t)$$

$$\frac{d P_n(t)}{dt} = (N - n + 1) \lambda P_{n-1}(t) - [(N - n) \lambda + n \mu] P_n(t) + (n+1) \mu P_{n+1}(t)$$

dacă $1 \leqslant n < s$

$$\frac{d P_n(t)}{dt} = (N - n + 1) \lambda P_{n-1}(t) - [(N - n) \lambda + s \mu] P_n(t) + s \mu P_{n+1}(t)$$

dacă $s \leqslant n < N$

$$\frac{d P_N(t)}{dt} = \lambda P_{N-1}(t) - s \mu P_N(t).$$

Vom determina probabilităţile stărilor în cazul regimului staţionar, adică atunci cînd :

$$\lim_{t \to \infty} \frac{d P_n(t)}{dt} = 0.$$

În acest caz sistemul de ecuaţii diferenţiale devine un sistem de ecuaţii algebrice :

$$\begin{cases} - N \lambda p_0 + \mu p_1 = 0 \\ (N - n + 1) \lambda p_{n-1} - [(N - n) \lambda + n\mu] p_n + (n + 1) \mu p_{n+1} = 0 \\ \qquad\qquad \text{dacă } 1 \leqslant n < s \\ (N - n + 1) \lambda p_{n-1} - [N - n) \lambda + s \mu] p_n + s \mu p_{n+1} = 0 \\ \qquad\qquad \text{dacă } s \leqslant n < N \\ \lambda p_{N-1} - s \mu p_N = 0 \end{cases}$$

Pentru $1 \leqslant n < s$ relaţiile mai pot fi scrise :

$$(N - n + 1) \lambda p_{n-1} - n \mu p_n = (N - n) p_n - (n + 1) \mu p_{n+1}$$

Scanned with OKEN Scanner

adică $z_n = z_{n+1}$ dacă notăm

$$z_n = (N - n + 1)\,\lambda\, p_{n-1} - n\,\mu\, p_n$$

Cum $z_1 = 0$ obținem $z_n = 0, \quad n = 1, 2, \ldots s$
și de aici

$$p_n = \frac{N - n + 1}{n}\,\rho\, p_{n-1} = \rho^n \frac{N(N-1)\cdots(N-n+1)}{n!}\, p_0 = C_N^n\, \rho^n p_0$$

Pentru $n = s, \quad s + 1, \ldots N - 1$ obținem

$$p_{s+1} = \frac{(N-s)\rho + s}{s}\, p_s - \frac{N - s + 1}{s}\,\rho\, p_{s-1} = \frac{N-s}{s}\,\rho\, p_s$$

. . . . . .

$$p_{s+r+1} = \left(\frac{\rho}{s}\right)^{r+1} \rho^s \frac{N(N-1)\cdots(N-s-r)}{s!}\, p_0$$

$$r = 0,1,\ldots, \quad N - s + 1$$

Rezultă așadar :

$$p_n = \begin{cases} C_N^n\, \rho^n p_0, & \text{dacă } 0 \leqslant n < s \\ \dfrac{N(N-1)\cdots(N-n+1)}{s^{n-s}\cdot s!}\,\rho^n\, p_0, & \text{dacă } s \leqslant n \leqslant N \end{cases}$$

unde p_0 se determină din relația $\sum_{n=0}^{N} p_n = 1$ adică

$$p_0 = \frac{1}{\sum_{n=s}^{s-1} C_N^n \rho^n + \sum_{n=s}^{N} \rho^n \dfrac{N(N-1)\cdots(N-n+1)}{s^{n-s}\cdot s!}}$$

Avînd probabilitățile stărilor putem calcula principalele caracteristici numerice ale modelului. Vom exprima mai întîi variabilele aleatoare și repartițiile lor și după aceea vom căuta să calculăm efectiv valorile lor medii.

Dacă notăm ca și pînă acum prin **Z** variabila aleatoare număr de unități în sistem, atunci :

$$Z : \begin{pmatrix} 0 & 1 & 2 \ldots N \\ p_0 & p_1 & p_2 \cdots p_N \end{pmatrix}$$

de unde

$$\bar{Z} = M(Z) = \sum_{n=0}^{N} n\, p_n.$$

Scanned with OKEN Scanner

Numărul de unităţi în şirul de aşteptare este o variabilă aleatoare **Q** cu repartiţia :

$$\mathbf{Q}:\begin{pmatrix} 0 & 1 & 2 \ldots \mathrm{N}-s \\ \sum_{n=0}^{s} p_n & p_{s+1} & p_{s+2} \cdots p_{\mathrm{N}} \end{pmatrix}$$

şi de aici

$$\bar{\mathbf{Q}} = \mathbf{M}(\mathbf{Q}) = \sum_{n=s+1}^{\mathrm{N}} (n-s)\, p_n$$

Numărul de staţii inactive, **L** are repartiţia

$$\mathbf{L}:\begin{pmatrix} 0 & 1 & 2 & \ldots s \\ \sum_{n=s}^{\mathrm{N}} p_n & p_{s-1} & p_{s-2} \cdots p_0 \end{pmatrix}$$

şi valoarea medie

$$\bar{\mathbf{L}} = \mathbf{M}(\mathbf{L}) = \sum_{n=0}^{s-1} (s-n)\, p_n$$

Numărul de staţii ocupate **O** are repartiţia

$$\mathbf{O}:\begin{pmatrix} 0 & 1 & 2 \ldots s & \\ p_0 & p_1 & p_2 \cdots \sum_{n=s}^{\mathrm{N}} p_n & \end{pmatrix}$$

$$\bar{\mathbf{O}} = \mathbf{M}(\mathbf{O}) = \sum_{n=0}^{s-1} n\, p_n + s \sum_{n=s}^{\mathrm{N}} p_n.$$

Timpul mediu de aşteptare în şir

$$\mathbf{W} = \frac{\bar{\mathbf{Q}}}{\mu\,(\mathrm{N} - \bar{\mathbf{L}})}$$

În fine să exprimăm şi probabilitatea ca o unitate să aştepte pentru a fi servită :

$$p(>0) = \sum_{n=s}^{\mathrm{N}} p_n.$$

Pentru calculul efectiv al acestor caracteristici precum şi a altora — vom folosi o metodă datorată lui Palm.

Scanned with OKEN Scanner

Mai întîi să exprimăm probabilitatea p_0. Egalitatea

$$p_0\left[\sum_{n=0}^{s-1} C_N^n \rho^n + \sum_{n=s}^{N} \rho^n \frac{N!}{(N-n)!\, s!\, s^{n-s}}\right] = 1$$

mai poate fi scrisă :

$$p_0\left[\sum_{n=0}^{s-1} C_N^n \rho^n + \frac{N!\,\rho^s}{s!\,(N-s)!}\sum_{m=0}^{N-s}\frac{(N-s)!}{(N-m-s)!}\left(\frac{\rho}{s}\right)^m\right] = 1$$

Să notăm

$$F\left(\frac{\rho}{s};\ N-s\right) = \sum_{m=0}^{N-s}\frac{(N-s)!}{(N-m-s)!}\left(\frac{\rho}{s}\right)^m$$

Dacă punem acum

$$P(X=k) = p\ (k,\ \theta) = e^{-\theta}\frac{\theta^k}{k!},\ k = 0,\ 1,\ 2,\ldots$$

și $N-s=k$; $\theta = \dfrac{s}{\rho}$, putem exprima $F\left(\dfrac{\rho}{s};\ N-s\right)$ în felul următor :

$$F\left(\frac{\rho}{s};\ N-s\right) = F\left(\frac{1}{\theta};\ k\right) = \frac{\sum_{m=0}^{k} p(k-m;\theta)}{p\,(k;\theta)} = \frac{1-P\,(k+1,\theta)}{p\,(k,\theta)}$$

unde

$$P\,(k+1,\ \theta) = \sum_{l=k+1}^{\infty} p(l,\ \theta).$$

Cu aceste precizări obținem :

$$p_0 = \frac{1}{\sum_{n=0}^{s-1} C_N^n \rho^n + C_N^s \rho^s \dfrac{1-P(k+1,\theta)}{p\,(k,\theta)}} =$$

$$= \frac{1}{\dfrac{1}{p\,(N,\theta)}\left[\sum_{n=0}^{s-1}\dfrac{s^n}{n!}e^{-\theta}\dfrac{\theta^{N-n}}{(N-n)!} + \dfrac{s^s}{s!}\{1-P\,(k+1,\theta)\}\right]}$$

sau încă

$$p_0 = \frac{p\,(N,\theta)}{S\,(N,s,\theta)}$$

Scanned with OKEN Scanner

unde am pus :

$$S(N, s, \theta) = \sum_{n=0}^{s-1} \frac{s^n}{n!} p(N - n, \theta) + \frac{s^s}{s!} \{1 - P(k+1, \theta)\}$$

Putem exprima corespunzător și probabilitățile p_n după cum urmează :

$$p_n = \begin{cases} \dfrac{s^n}{n!} \cdot \dfrac{p(N-n,\theta)}{S(N, s, \theta)} & , \quad 0 \leqslant n < s \\ \dfrac{s^s}{s!} \cdot \dfrac{p(N-n, \theta)}{S(N, s, \theta)} & , \quad s \leqslant n \leqslant N. \end{cases}$$

Să aflăm mai întîi $\bar{O} = M(O)$.
Conform definiției :

$$\bar{O} = M(O) = \sum_{n=0}^{s-1} n\, p_n + s \sum_{n=s}^{N} p_n = \sum_{n=0}^{s-1} n \cdot \frac{s^n}{n!} \cdot \frac{p(N-n,\theta)}{S(N, s, \theta)} +$$

$$+ s \sum_{n=1}^{N} \frac{s^s}{s!} \frac{p(N-n, \theta)}{S(N, s, \theta)} = \frac{s}{S(N, s, \theta)} \left[\sum_{n=1}^{s-1} \frac{s^{n-1}}{(n-1)!} p(N - n, \theta) + \right.$$

$$\left. + \frac{s^s}{s!} \{1 - P(N - s + 1, \theta)\} \right] = \frac{s\, S(N-1, s, \theta)}{S(N, s, \theta)}.$$

Pentru ușurința calculelor să mai punem în evidență variabila aleatoare, numărul de unități existente în afara sistemului, U. Atunci :

$$U : \begin{pmatrix} 0 & 1 & 2 & \dots & N \\ p_N & p_{N-1} & p_{N-2} & \dots & p_0 \end{pmatrix}$$

și $\bar{U} = M(U) = \sum_{n=0}^{N} (N - n)\, p_n$

Se constată imediat însă, că

$$\frac{M(U)}{M(O)} = \frac{0 \cdot p_N + 1 \cdot p_{N-1} + \dots + N p_0}{0 \cdot p_0 + 1 \cdot p_1 + \dots + (s-1)\, p_{s-1} + s \sum_{n=s}^{N} p_n} = \frac{1}{\rho}.$$

$$\text{Deci } M(U) = \frac{M(O)}{\rho} = \frac{s\, S(N-1, s, \theta)}{\rho\, S(N, s, \theta)}$$

Scanned with OKEN Scanner

Cum $\mathbf{Z} = \mathbf{N} - \mathbf{U}$ rezultă :

$$\bar{\mathbf{Z}} = \mathbf{M}(\mathbf{Z}) = \mathbf{N} - \mathbf{M}(\mathbf{U}) = \mathbf{N} - \frac{s\,\mathbf{S}(\mathbf{N}-1, s, \theta)}{\rho\,\mathbf{S}(\mathbf{N}, s, \theta)}$$

Pentru a calcula valoarea medie $\bar{\mathbf{Q}} = \mathbf{M}(\mathbf{Q})$ observăm că

$$\mathbf{Q} + \mathbf{O} + \mathbf{U} = \mathbf{N}$$

și deci

$$\bar{\mathbf{Q}} = \mathbf{M}(\mathbf{Q}) = \mathbf{N} - \mathbf{M}(\mathbf{O}) - \mathbf{M}(\mathbf{U}) =$$

$$= \mathbf{N} - \frac{s\,\mathbf{S}(\mathbf{N}-1, s, \theta)}{\mathbf{S}(\mathbf{N}, s, \theta)} - \frac{s\,\mathbf{S}(\mathbf{N}-1, s, \theta)}{\rho\,\mathbf{S}(\mathbf{N}, s, \theta)} =$$

$$= \mathbf{N} - s\,\frac{\rho+1}{\rho} \cdot \frac{\mathbf{S}(\mathbf{N}-1, s, \theta)}{\mathbf{S}(\mathbf{N}, s, \theta)}$$

Timpul mediu de așteptare în șir

$$\mathbf{W} = \frac{\bar{\mathbf{Q}}}{\mu(\mathbf{N} - \bar{\mathbf{L}})}$$

îl calculăm imediat ce exprimăm $\bar{\mathbf{L}}$.

Dar $\mathbf{L} + \mathbf{O} = s$ și de aici :

$$\bar{\mathbf{L}} = s - \bar{\mathbf{O}} = s - \frac{s\,\mathbf{S}(\mathbf{N}-1, s, \theta)}{\mathbf{S}(\mathbf{N}, s, \theta)} = s\left(1 - \frac{\mathbf{S}(\mathbf{N}-1, s, \theta)}{\mathbf{S}(\mathbf{N}, s, \theta)}\right)$$

Deci :

$$\mathbf{W} = \frac{\mathbf{N} - s\,\dfrac{\rho+1}{\rho} \cdot \dfrac{\mathbf{S}(\mathbf{N}-1, s, \theta)}{\mathbf{S}(\mathbf{N}, s, \theta)}}{\mu\left[\mathbf{N} - s\left(1 - \dfrac{\mathbf{S}(\mathbf{N}-1, s, \theta)}{\mathbf{S}(\mathbf{N}, s, \theta)}\right)\right]}$$

În fine probabilitatea ca o unitate să aștepte

$$p(>0) = \sum_{n=s}^{N} p_n = \frac{s^s}{s!} \sum_{n=s}^{N} \frac{p(\mathbf{N}-n, \theta)}{\mathbf{S}(\mathbf{N}, s, \theta)} =$$

$$= \frac{s^s}{s!}\,\frac{1 - \mathbf{P}(\mathbf{N}-s+1, \theta)}{p(\mathbf{N}, \theta) \cdot \mathbf{S}(\mathbf{N}, s, \theta)}$$

Scanned with OKEN Scanner

Exemplul 7.6. Într-un atelier sînt 20 de mașini. Defecțiunile la mașini sînt rare iar reparațiile se execută rapid. Costul unei ore de indisponibilitate a unei mașini este $C_1 = 2000$ unități monetare, iar un mecanic primește o retribuție orară de $C_2 = 8$ unități monetare. Care este numărul optim de mecanici pe care trebuie să-i încadreze întreprinderea la acest atelier astfel încît costul global să fie minim, știind că $\rho = 0,1$.

Soluție. Costul global $\mathbf{F}(s)$ se determină astfel:

$$\mathbf{F}(s) = C_2 \sum_{n=0}^{s-1} (s-n)\, p_n + C_1 \sum_{n=s+1}^{N-s} (n-s) p_n.$$

Utilizînd modelul de așteptare cu mai multe stații cu veniri dintr-o populație finită obținem pentru diferitele valori ale lui s probabilitățile p_n din tabelul de mai jos.

Tabelul 7.2

s	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8
3	0,136	0,272	0,258	0,155	0,088	0,047	0,023	0,011	0,005
4	0,146	0,292	0,278	0,166	0,071	0,028	0,010	0,003	0,001
5	0,148	0,296	0,281	0,168	0,071	0,022	0,006	0,001	0,000
6	0,148	0,297	0,282	0,169	0,072	0,023	0,006	0,001	0,000
7	0,148	0,297	0,282	0,169	0,072	0,023	0,006	0,001	0,000

Un calcul amănunțit a dovedit că se exclud valorile $s = 2$, $s = 8,9$. De aici se obține tabelul cheltuielilor globale pentru valorile lui s menționate mai sus, tabelul 7.3.

Tabelul 7.3

s	$\sum_{n=0}^{s-1} (s-n)\, p_n$	$\sum_{n=s+1}^{N-s} (n-s)\, p_n$	$\mathbf{F}(s) = C_2 \sum_{n=0}^{s-1} (s-n)\, p_n + C_1 \sum_{n=s+1}^{N-s} (n-s)\, p_n$
3	0,32	1,20	$2000 \cdot 0,32 + 8 \cdot 1,20 = 649,6$
4	0,06	2,18	$2000 \cdot 0,06 + 8 \cdot 2,18 = 137,44$
5	0,01	3,17	$2000 \cdot 0,01 + 8 \cdot 3,17 = 45,36$
6	—	4,17	$8 \cdot 4,17 = 33,36$
7	—	5,16	$8 \cdot 5,16 = 41,28$

Deci cheltuielile globale minime se obțin pentru $s = 6$ și prin urmare întreprinderea va angaja 6 mecanici pentru repararea mașinilor.

Scanned with OKEN Scanner

Capitolul VIII

TEORIA JOCURILOR ŞI A DECIZIEI

8.1. Elemente privind jocurile strategice

Jocul este un proces competitiv care se desfăşoară între mai mulţi participanţi numiţi jucători, dintre care cel puţin unul este inteligent şi prudent, adică poate analiza situaţia şi hotărî asupra acţiunilor viitoare.

Jocurile cu doi participanţi au o deosebită importanţă în modelarea proceselor economice.

Partida este desfăşurarea acţiunilor jucătorilor, după anumite reguli. Orice partidă are o stare iniţială şi o stare finală, aceasta din urmă determinînd, pe baza regulilor jocurilor, un cîştig sau o pierdere pentru fiecare jucător.

Strategia este o colecţie de succesiuni de acţiuni ale unui jucător, fiecare dintre succesiuni fiind pregătită ca o reacţie faţă de strategia adversarului (care poate fi, uneori, natura), în scopul atingerii obiectivului propus, adică a acelei stări finale căreia regulile jocului îi asociază maximum de cîştig posibil.

O reprezentare comodă a unui joc cu doi jucători este cea matriceală (tabelul 8.1).

Tabelul 8.1

A \ B	B_1	B_2	...	B_n
A_1	a_{11}	a_{12}	...	a_{1n}
A_2	a_{21}	a_{22}	...	a_{2n}
.	.	.		.
.	.	.		.
.	.	.		.
A_m	a_{m1}	a_{m2}	...	a_{mn}

În acest tabel, **A** reprezintă unul dintre jucători, iar **B** reprezintă adversarul (care poate fi uneori „natura”);

$\mathbf{A} = \{A_1, A_2, \ldots, A_m\}$ este mulţimea strategiilor lui **A**,

$\mathbf{B} = \{B_1, B_2, \ldots, B_n\}$ mulţimea strategiilor lui **B**, iar $a_{ij} (i = 1,2, \ldots, m;$

Scanned with OKEN Scanner

$j = 1, 2, \ldots, n$) reprezintă *consecința* corespunzătoare adoptării strategiei $\mathbf{A}_i$, de către **A** și a strategiei $\mathbf{B}_j$ de către **B**.

În procesul de decizie pe care-l reprezintă alegerea uneia sau alteia dintre strategii, ar trebui luate în considerație *utilitățile** $u(a_{ij})$ ale fiecărei consecințe. Pentru simplificare, se va considera însă, convențional, că toate consecințele sînt valori bănești sau că ele reprezintă chiar utilități.

O primă categorie de jocuri sînt așa-numitele jocuri cu punct șa, caracterizate prin aceea că un raționament corect impune fiecăruia dintre cei doi jucători alegerea cîte unei anumite strategii optime. Ansamblul celor două strategii optime constituie o soluție a jocului. Un exemplu de calcul va contribui la fixarea ideilor.

Fie jocul reprezentat prin matricea de mai jos (tabelul 8.2).

Tabelul 8.2

Strategii ale lui A \ Strategii ale lui B	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{B}_3$
$\mathbf{A}_1$	*1*	*−1*	*−1,5*
$\mathbf{A}_2$	*0,5*	*−0,5*	*0*

(Tabelul 8.2 se interpretează astfel: dacă jucătorul **A** alege strategia $\mathbf{A}_1$, iar jucătorul **B** strategia $\mathbf{B}_1$, jucătorul **A** cîștigă *1* punct, iar **B**, pierde *1*; dacă **A** alege strategia $\mathbf{A}_1$, iar **B** strategia $\mathbf{B}_2$, jucătorul **A** pierde *1* punct, iar **B** cîștigă *1* punct ș.a.m.d.).

Cum va proceda jucătorul **A** pentru a alege strategia optimă?

Alegerea strategiei $\mathbf{A}_1$ poate conduce la cel mai mare cîștig pentru **A**, în cazul cînd **B** alege strategia $\mathbf{B}_1$, dar și la cea mai mare pierdere, în cazul cînd **B** alege strategia $\mathbf{B}_3$. În schimb strategia $\mathbf{A}_2$ este mai prudentă, neputînd, ce e drept, aduce decît un cîștig maxim de *0,5*, dar, în cazul cel mai nefavorabil, nici pierderea nu e mai mare decît *0,5*. Se observă de asemenea, că atît pentru strategia $\mathbf{B}_2$ cît și pentru $\mathbf{B}_3$, ambele consecințe sînt mai avantajoase decît consecințele corespunzătoare ale strategiei $\mathbf{B}_1$. În terminologia teoriei jocurilor se va spune că strategiile $\mathbf{B}_2$ și $\mathbf{B}_3$ domină strategia $\mathbf{B}_1$, care poate fi în consecință eliminată ca fiind totdeauna dezavantajoasă pentru jucătorul **B**. Matricea, modificată, va avea următoarea înfățișare (tabelul 8.3):

Tabelul 8.3

Strategii ale lui A \ Strategii ale lui B	$\mathbf{B}_2$	$\mathbf{B}_3$
$\mathbf{A}_1$	*−1*	*−1,5*
$\mathbf{A}_2$	*−0,5*	*0*

* Conceptul de utilitate va fi definit în paragraful 8.2.

Scanned with OKEN Scanner

În matricea astfel redusă se observă că strategia A_2 domină pe A_1 și, în consecință, eliminînd-o pe aceasta din urmă, matricea capătă următoarea înfățișare finală:

Tabelul 8.4

Strategii ale lui B / Strategii ale lui A	B_2	B_3
A_2	$-0,5$	0

Este evident că jucătorul **B** va alege strategia B_2. Așadar, printr-un șir de raționamente ambii jucători și-au ales strategiile: jucătorul **A** strategia A_2, iar jucătorul **B** strategia B_2. Conform acestor alegeri jucătorul **A** va pierde *0,5* puncte, iar jucătorul **B** va cîștiga *0,5* puncte. Este important de subliniat că aceste cifre constituie valori minime la care se pot aștepta cei doi jucători alegînd cele două strategii, și anume, valori care rezultă cînd adversarul „joacă" corect. Se poate vedea cu ușurință că dacă unul dintre jucători greșește, adică alege altă strategie, celălalt jucător, dacă „joacă" corect, cîștigă mai mult decît valorile indicate mai sus.

După cum s-a observat, la jocurile de tipul de mai sus raționamentul impune fiecărui jucător cîte o strategie optimă numită *strategie pură*.

Perechea celor două strategii optime constituie o soluție a jocului și determină un așa-numit *punct șa*. Cîștigul, respectiv pierderea, înregistrată de cei doi jucători (în cazul de mai sus *0,5*), ca rezultat al aplicării perechii de strategii optime, reprezintă *valoarea jocului*.

Determinarea strategiilor optime la jocurile avînd punct șa se poate face fără a relua întregul raționament, pe baza așa-numitului *principiu maximin*.

Fie un joc de ordinul $m \times n$, avînd următoarea matrice asociată:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ a_{m1} & \dots & a_{mn} \end{bmatrix}$$

Conform principiului maximin, primul jucător va alege acea strategie căreia îi corespunde un cîștig de minimum:

$$v_1 = \max_i (\min_j a_{ij}); \quad 1 \leqslant i \leqslant m; \quad 1 \leqslant j \leqslant n.$$

Pentru determinarea valorii v_1 și a strategiei corespunzătoare el va proceda astfel: va determina toate valorile minime pe linii ($\min a_{ij}$) și dintre acestea va lua valoarea maximă (maximin a_{ij}).

Cel de-al doilea jucător va proceda asemănător alegînd toate maximele pe coloană ($\max a_{ij}$) și luînd pe cel mai mic dintre ele (minmax a_{ij}):

$$v_2 = \min_j (\max_i a_{ij}) \quad 1 \leqslant j \leqslant n; \quad 1 \leqslant i \leqslant m$$

Scanned with OKEN Scanner

În cazul jocurilor avînd punct șa cele două valori sînt egale și corespund valorii jocului :

$$v = v_1 = v_2 = \max_i (\min_j a_{ij}) = \min_j (\max_i a_{ij}); \quad 1 \leqslant i \leqslant m; \quad 1 \leqslant j \leqslant n.$$

O a doua categorie o constituie jocurile fără punct șa.

Acestea se caracterizează prin faptul că un raționament oricît de riguros al jucătorilor nu-i va conduce în mod necesar la alegerea unei anumite perechi de strategii, ca în cazul jocurilor cu punct șa.

În cazul cînd procesul economic al cărui model matematic este un joc fără punct șa și se repetă, în condiții identice de multe ori, există o soluție specifică a problemei constînd în determinarea strategiilor mixte optime ale celor doi parteneri.

O strategie mixtă a jucătorului **A**, care are la dispoziție m strategii : $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_m$, este un șir de m numere $x_1, x_2, \ldots, x_m$ ($x_1 + x_2 + \ldots + x_m = 1$; $x_1, x_2, \ldots, x_m \geqslant 0$), care reprezintă probabilitățile de a folosi fiecare din cele m strategii, în desfășurarea unui număr mare de partide. Un înțeles similar are noțiunea de strategie mixtă a jucătorului **B**. Se demonstrează în teoria jocurilor că există o pereche de strategii mixte ale celor doi jucători, care sînt optime, asigurînd un cîștig mediu maxim, respectiv o pierdere medie minimă, egale între ele.

Cu ajutorul unor jocuri avînd doi adversari care raționează, se modelează situații conflictuale între două sau mai multe părți (acestea din urmă se reduc tot la jocuri cu doi participanți, prin formarea așa-numitelor „coaliții"), fiecare dintre părți urmărind alegerea unei strategii care să-i asigure un rezultat cît mai avantajos în detrimentul adversarului sau adversarilor.

8.2. Teoria deciziei

Teoria deciziei are drept obiectiv modelarea structurii generale a procesului decizional. Deși este de obicei considerată o disciplină independentă, avînd în vedere rolul ei în modelarea unor procese decizionale care apar în activitatea organizatorică și de conducere, îi rezervăm acest paragraf în prezentul curs consacrat cercetării operaționale.

În cadrul teoriei deciziei, problemele studiate pot fi privite în perspectivă descriptivă, cînd se urmărește înțelegerea și descrierea mecanismului decizional, sau în perspectiva normativă, cînd se caută formularea unor reguli raționale și eficiente de urmat în activitatea decizională. Adesea, cele două perspective coexistă, completîndu-se reciproc.

Elementele procesului decizional

În general, la un proces de decizie se pot distinge următoarele elemente :

a) decidentul, adică individul sau mulțimea de indivizi care urmează să ia decizia ;

b) formularea (denumirea) pe care o dă decidentul problemei decizionale întîlnite ;

Scanned with OKEN Scanner

c) mulţimea variantelor (alternativelor) posibile, care caracterizează o situaţie decizională şi din care trebuie aleasă o variantă optimă;

d) mulţimea consecinţelor anticipate pentru fiecare variantă luată în considerare;

e) mulţimea criteriilor de decizie ale decidentului, adică punctele de vedere pe care le ia în considerare la alegerea variantei optime;

f) obiectivele (scopurile) decidentului, respectiv consecinţele propuse a fi atinse pentru criteriile de decizie alese;

g) stările naturii.

Schematic elementele procesului decizional pot fi reprezentate astfel:

Tabelul 8.5

Variante	Starea naturii N_1				Starea naturii N_2				...	Starea naturii N_n		
	Criteriul X_1	Criteriul X_2	...	Criteriul X_n	Criteriul X_1	Criteriul X_2	...	Criteriul X_n		Criteriul X_1	Criteriul X_2	Criteriul X_n
V_1	x_{111}	x_{112}	...	x_{11r}	x_{121}	x_{122}	...	x_{12r}		x_{1n1}	x_{1n2}	x_{1nr}
V_2	x_{211}	x_{212}	...	x_{21r}	x_{221}	x_{223}	...	x_{22r}		x_{2n1}	x_{2n2}	x_{2n2}
.	.	.		.	.	.		.	...	.	.	.
.	.	.		.	.	.		.		.	.	.
.	.	.		.	.	.		.		.	.	.
V_m	x_{m11}	x_{m22}	...	x_{m1r}	x_{m21}	x_{m23}	...	x_{m2r}		x_{mn1}	x_{mn2}	x_{mnr}

În reprezentarea de mai sus consecinţele s-au notat cu x_{ijk} ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$; $k = 1, 2, \ldots, r$) iar decidentul, formularea problemei şi obiectivele urmărite trebuie specificate separat.

Exemplul 8.1. Un grup de specialişti trebuie să aleagă o soluţie privind dotarea unei întreprinderi cu echipament electronic de calcul. Elementele problemei sînt cele din tabelul de mai jos:

Tabelul 8.6

Variante	Starea naturii N_1 (favorabilă)			Starea naturii N_2 (nefavorabilă)		
	Criteriul cost	Criteriul durată de livrare	Criteriul calitate software	Criteriul cost	Criteriul durată de livrare	Criteriul calitate software
V_1	*4* milioane	*9* luni	bună	*4,4* mil.	*1* an	bună
V_2	*4,8* "	*6* luni	f. bună	*4,8* mil.	*8* luni	f. bună
V_3	*4,5* "	*8* luni	excelentă	*5* mil.	*10* luni	excelentă

Stările naturii luate în considerare pot fi interpretate ca nişte condiţii posibile de conjunctură pe piaţa internaţională a calculatoarelor.

Obiectivele urmărite de decident trebuie specificate pentru fiecare criteriu în parte şi ele pot fi, de exemplu: pentru criteriul cost — minimiza-

Scanned with OKEN Scanner

rea costului, pentru criteriul durată — nedepășirea termenului de livrare de 8 luni, iar pentru criteriul software — calitate cît mai bună.

Conceptul de utilitate

Conceptul de utilitate apare în teoria deciziei ca urmare a necesității de a compara între ele variante decizionale caracterizate prin mai multe consecințe. Dacă ne referim la exemplul de mai sus și ținem seama, pentru simplificare, numai de starea naturii N_1, variantele care trebuie comparate în vederea alegerii celei mai avantajoase sînt caracterizate din punctul de vedere al consecințelor în felul următor:

V_1 (cost *4* milioane; durata de livrare *9* luni; calitate software bună);

V_2 (cost *4,8* milioane, durata de livrare *6* luni, calitate software foarte bună),

V_3 (cost *4,5* milioane, durata de livrare *8* luni, calitate software excelentă).

Este evident că în urma comparației celor trei variante nu ne putem pronunța care este mai avantajoasă. Este firesc să urmărim alegerea variantei care ne este cea mai folositoare, deci care are utilitatea maximă. Pentru aceasta este necesar să se definească riguros utilitatea. Vom prezenta, în continuare, principalele idei care stau la baza definiției dată utilității de J. von Neumann și O. Morgenstern denumită în continuare axiomatica V. N—M, cea mai cunoscută și întrebuințată definiție.

Utilitatea este o mărime subiectivă, depinzînd de aprecierea decidentului. Pentru a îngrădi caracterul subiectiv al estimării utilității se introduc cinci axiome:

Axioma 1. Două variante decizionale V_i și V_j pot fi totdeauna comparate între ele, decidentul putînd pronunța una din următoarele opțiuni:

— preferă pe V_i lui V_j ($V_i \succ V_j$);

— preferă pe V_j lui V_j ($V_j \succ V_i$);

— cele două variante îi sînt indiferente ($V_i \sim V_j$)

Axioma 2. Relația de preferință este tranzitivă, iar relația de indiferență este tranzitivă și simetrică.

Axioma 3. În afara mulțimii $V = \{V_1, V_2, \ldots, V_m\}$ a variantelor simple, decidentul poate lua în considerare un tip special de variante simple numite „mixturi probabilistice", de tipul: $V' = [p\,V_i,\ (1-p)V_j]$, în care p este probabilitatea realizării variantei V_i iar $(1-p)$ — probabilitatea realizării variantei V_j.

Axioma 4. Fiind date trei variante V_i, V_j și V_k și un decident care exprimă relația $V_i \succ V_j \succ V_k$, există o mixtură

$$V' = [p'V_i,\ (1-p')V_k], \text{ astfel ca } V' \succ V_j$$

și o altă mixtură

$$V'' = [p''V_i, (1-p'')V_k], \text{ astfel ca } V_j \succ V''.$$

Axioma 5. Dacă o variantă (V_i) este preferată altei variante (V_j) atunci o mixtură $[pV_i,\ (1-p)\,V_k]$ va fi totdeauna preferată mixturii $[pV_j, (1-p)\,V_k]$.

Scanned with OKEN Scanner

Pe baza acestor axiome se introduce funcţia de utilitate $u(\mathbf{V}_i)$, avînd ca domeniu de definiţie mulţimea variantelor, iar valori în mulţimea numerelor reale. Principalele sale proprietăţi sînt următoarele :

a. Fie $\mathbf{V}_i$ şi $\mathbf{V}_j$ două variante decizionale, atunci $\mathbf{V}_i \succ \mathbf{V}_j$ dacă şi numai dacă $u(\mathbf{V}_i) > u(\mathbf{V}_j)$;

b. $u[p\mathbf{V}_i,\ (1-p)\,\mathbf{V}_j] = p \cdot u(\mathbf{V}_i) + (1-p) \cdot u(\mathbf{V}_j)$;

c. Dacă funcţia de utilitate posedă proprietăţile *a* şi *b*, atunci ea poate suferi o transformare liniară pozitivă.

$$u'(\mathbf{V}_i) = a \cdot u(\mathbf{V}_i) + b, \text{ cu } a > 0 \text{ şi } b \geqslant 0.$$

Procedeul practic de estimare a utilităţilor constă în a considera cunoscută utilitatea a două variante aparţinînd mulţimii $\mathbf{V} = \{\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_m\}$, utilitatea celorlalte variante determinîndu-se cu ajutorul proprietăţii *b*.

Exemplul 8.2. Pentru variantele $\mathbf{V}_1$, $\mathbf{V}_2$ şi $\mathbf{V}_3$ specificate mai sus stabilim următoarea ordine de preferinţă : $\mathbf{V}_3 \succ \mathbf{V}_1 \succ \mathbf{V}_2$. Apreciem * $u(\mathbf{V}_3) = 1$ şi $u(\mathbf{V}_2) = 0$ iar pentru determinarea utilităţii lui $\mathbf{V}_1$ construim mixtura $[p\mathbf{V}_3 ;\ (1-p)\mathbf{V}_2]$ şi estimăm probabilitatea ** p pentru care vom avea :

$$\mathbf{V}_1 \sim [p\mathbf{V}_3 ;\ (1-p)\mathbf{V}_2]$$

Să presupunem că se apreciază $p = 0{,}6$ atunci conform proprietăţii *b* se scrie :

$$u(\mathbf{V}_1) = u[p\mathbf{V}_3 ;\ (1-p)\,\mathbf{V}_2] = 0{,}6 \cdot \mu(\mathbf{V}_3) + 0{,}4 \cdot \mu(\mathbf{V}_2) = 0{,}6.$$

Tipuri de procese de decizie şi algoritmi de rezolvare

1. *Clasificări ale proceselor decizionale.* Se pot face clasificări ale proceselor de decizie din mai multe puncte de vedere :

a) după numărul variantelor posibile :
— procese de decizie cu număr finit de variante,
— procese de decizie cu număr infinit *** de variante,

b) după numărul criteriilor luate în considerare :
— procese de decizie cu un singur criteriu,
— procese de decizie cu mai multe criterii (multidimensionale),

c) după numărul de stări ale naturii şi probabilitatea de realizare a acestora :
— procese de decizie în condiţii de certitudine (o singură stare a naturii, probabilitatea de realizare *1*),
— procese de decizie în condiţii de risc (mai multe stări ale naturii cu probabilităţi de realizare cunoscute).

---

* S-a luat $u_{max} = 1$ şi $u_{min} = 0$, dar axiomatica $v \cdot \mathbf{N} - \mathbf{M}$ permite folosirea oricărei scări, cu condiţia ca utilităţile să fie numere reale.

** p este o probabilitate subiectivă estimată de decident.

*** Un exemplu de astfel de proces decizional este o problemă de programare liniară, în care mulţimea variantelor posibile este echivalentă cu mulţimea soluţiilor sistemului de restricţii.

Scanned with OKEN Scanner

— procese de decizie în condiții de incertitudine (mai multe stări ale naturii, fără cunoașterea probabilităților de realizare a lor).

d) după numărul de procese decizionale considerate în succesiune:
— proces decizional unic (desfășurat la momentul t),
— succesiuni de procese decizionale desfășurate la momente diferite.

e) după numărul decidenților:
— procese decizionale cu decident unic,
— decizii de grup.

În continuare vom prezenta, în perspectiva clasificărilor de mai sus, diferite tipuri de procese de decizie.

2. *Procese de decizie multidimensionale.* După cum s-a arătat, atunci cînd estimăm utilitatea unor variante decizionale caracterizate prin mai multe consecințe, luăm de fapt în considerare mai multe criterii decizionale. Estimarea utilității este însă globală, adică nu se analizează separat, consecință cu consecință. Spre deosebire de acest procedeu de comparare a utilității consecințelor, care are originea în teoriile economice ale secolului trecut, teoria modernă a deciziei a simțit nevoia să introducă estimarea utilităților separat pe criterii.

Fie deci procesul decizional multidimensional:

Tabelul 8.7

Criterii / Variante	X_1	X_2	...	X_r
V_1	x_{11}	x_{12}	...	x_{1r}
V_2	x_{21}	x_{22}	...	x_{2r}
.	.	.		.
.	.	.		.
.	.	.		.
V_m	x_{m1}	x_{m2}	...	x_{mr}

Utilitatea unei consecințe V_i va fi:

$$u(V_i) = u(x_{i1}) + u(x_{i2}) + \ldots + u(x_{ir}).$$

Principalele probleme pe care le ridică teoria deciziilor multidimensionale sînt:

1) cum estimăm utilitatea consecințelor?

2) în ce condiții utilitățile consecințelor sînt aditive?

Dintre procedeele propuse pentru estimarea utilităților consecințelor la acest tip de proces decizional, cel mai avantajos pare a fi acela în care se consideră mulțimea tuturor consecințelor $X = \{x_{11}, x_{12}, \ldots, x_{m1}, x_{12}, x_{22}, \ldots, x_{m2}, \ldots, x_{1r}, x_{2r}, \ldots, x_{mr}\}$ și se aplică metoda Neumann-Morgenstern tuturor elementelor mulțimii.

În ceea ce privește aditivitatea multidimensională a utilităților, problema este foarte spinoasă, nu există încă procedee unanim acceptate de

Scanned with OKEN Scanner

operare. De regulă se acceptă posibilitatea adunării utilităților atunci cînd criteriile sînt independente * .

Considerînd procesul de decizie multidimensională din tabelul 8.6 și anume numai starea naturii N_1, facem estimările :

$u_{max} = u(6\ luni) = 1$

$u_{min} = u(4{,}8\ milioane) = 0.$

Utilitatea celorlalte consecințe se va determina cu ajutorul mixturilor, de exemplu :

$u(4\ milioane) = p \cdot u(6\ luni) + (1 - p) \cdot u(4{,}8\ milioane)$ și dacă estimăm $p = 0{,}9$, vom avea :

$u(4\ milioane) = 0{,}9.$

Tabelul 8.8

Criterii / Variante	Cost	Durată de livrare	Calitate software	Criteriu global
V_1	0,9	0,2	0,4	1,5
V_2	0	1	0,6	1,6
V_3	0,3	0,3	0,8	1,4

Presupunînd că aplicăm procedeul pentru toate consecințele și că determinăm astfel toate utilitățile, se va putea construi tabelul utilităților (tabelul 8.8).

Varianta optimă va fi cea de utilitate globală maximă adică V_2.

Tabelul 8.9

Stări ale naturii / Variante	N_1 p_1	N_2 p_2	...	N_n p_n
V_1	x_{11}	x_{12}	...	x_{1n}
V_2	x_{21}	x_{22}	...	x_{2n}
.	.	.		.
.	.	.		.
.	.	.		.
V_m	x_{m1}	x_{m2}	...	x_{mn}

3. *Procese de decizie în condiții de risc și incertitudine.*

a. Condiții de risc. Fie procesul de decizie cu mai multe stări ale naturii din tabelul 8.9. Să presupunem că pentru fiecare stare a naturii cunoaștem probabilitățile de realizare.

---

* Independența criteriilor are o semnificație specială, de esență logic-cauzală.

Scanned with OKEN Scanner

Pentru a determina varianta optimă în condiţii de risc se vor estima mai întîi toate utilităţile consecinţelor după metodele cunoscute.

Varianta optimă va fi aceea căreia îi corespunde utilitatea medie ponderată maximă.

$$\max_i \sum_{j=1}^{n} p_i \cdot u(x_{ij}) \quad \begin{matrix} i = 1,\ 2,\ \ldots,\ m \\ p_1 + p_2 + \ldots + p_n = 1. \end{matrix}$$

b. Condiţii de incertitudine. Fie problema de decizie definită mai sus, dar fără a cunoaşte probabilitatea $p_j (j = 1,2,\ldots, n)$ de realizare a stărilor naturii.

Găsirea soluţiei optime în condiţii de incertitudine poate fi făcută utilizînd una din regulile de mai jos * .

● *Regula prudentă (Wald).* Se alege varianta căreia îi corespunde utilitatea

$$\max_i \ [\min_j u(x_{ij})] \quad \begin{matrix} i = 1,2,\ldots\ m \\ j = 1,2,\ldots,\ n \end{matrix}$$

● *Regula regretului (Savage).* Se alege varianta căreia îi corespunde utilitatea :

$$\min_i \ [\max_j \ [(\max_i \ u(x_{ij}) - u(x_{ij})]] \quad \begin{matrix} i = 1,2,\ldots,\ m \\ j = 1,2,\ldots,\ n \end{matrix}$$

● *Regula lui Laplace.* Se consideră problema în condiţii de risc cu probabilităţile de realizare a stărilor naturii egale între ele.

Exemplul 8.3. Să reluăm problema de decizie a alegerii unui echipament electronic de calcul şi să considerăm că datele de care dispunem impun luarea în considerare a trei stări ale naturii. Să presupunem estimate utilităţile pe fiecare criteriu în parte şi însumate, ca utilităţi globale pe fiecare stare a naturii (tabelul 8.10).

Tabelul 8.10

Stări ale naturii / Variante	N_1	N_2	N_3
V_1	1,5	1,3	1,4
V_2	1,6	1,5	1,4
V_3	1,4	1,2	1,5

* În cazul în care unele consecinţe decizionale sînt foarte nefavorabile, avînd caracter dezastruos, se va aplica în mod obligatoriu regula prudentă.

Scanned with OKEN Scanner

● *Regula prudentă* $\max [\min (1,5; 1,3; 1,4), \min (1,6; 1,5; 1,4); \min (1,4; 1,2; 1,5)] = \max [1,3; 1,4; 1,2] = 1,4$ deci alegem V_2.

● *Regula regretului.* Se construiește așa-numita „matrice a regretelor" scăzînd pentru fiecare stare a naturii toate utilitățile din cea maximă :

0,1	0,2	0,1
0	0	0,1
0,2	0,3	0

Acestei matrice îi aplicăm regula prudentă ținînd seama că cele mai avantajoase sînt „regretele" mici :
$\min [\max (0,1; 0,2; 0,1), \max (0; 0; 0,1), \max (0,2; 0,3; 0] = \min [0,2; 0,1; 0,3] = 0,1$ rezultă deci tot V_2.

● *Regula lui Laplace* $\max \left[\frac{1}{3}(1,5 + 1,3 + 1,4); \frac{1}{3}(1,6 + 1,5 + 1,4); \frac{1}{3}(1,4 + 1,2 + 1,5)\right] = \max [1,4; 1,5; 1,37] = 1,5$ deci tot V_2^*.

4. *Succesiuni de procese de decizie, arbori decizionali.* Procesul de decizie în condiții de risc sau incertitudine, modelat schematic prin tabelul 8.6 și avînd regulile de rezolvare descrise mai sus, reprezintă în cele mai multe cazuri practice doar o etapă dintr-o succesiune de decizii în condiții de risc sau de incertitudine. Reprezentarea succesiunii proceselor de decizii se face prin așa-numitul „*arbore decizional*".

În figura 8.1 se prezintă arborele decizional pentru următoarea situație :

Trebuie aleasă o variantă de execuție pentru o investiție, din două variante posibile (V_1 și V_2). Execuția lucrării se poate desfășura în condiții favorabile în ceea ce privește condițiile de import ale unor utilaje (N_1^1) sau în condiții nefavorabile (N_2^1) influențînd evident costul.

După terminarea investiției, exploatarea ei se poate face în trei schimburi (V_1^2) sau numai în două (V_2^2), antrenînd, în funcție de conjunctura favorabilă (N_1^2) sau nefavorabilă (N_2^2) a pieței, beneficii sau pierderi la sfîrșitul unei perioade de exploatare (de exemplu 5 ani).

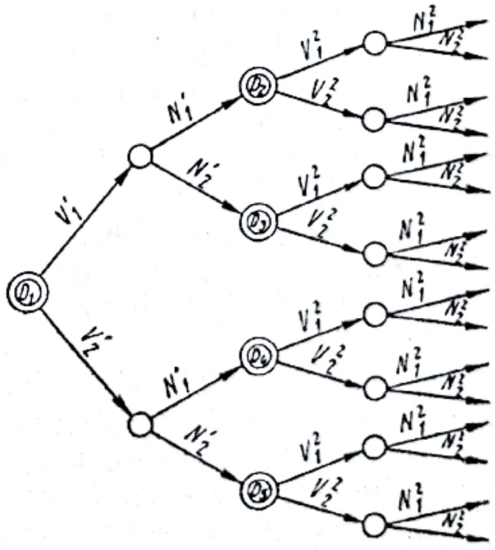


Fig. 8.1.

* Faptul că prin aplicarea celor trei reguli a rezultat aceeași variantă optimă este o întîmplare.

În arborele decizional din figura 8.1 sînt 8 strategii posibile:

1) $V_1^1\ N_1^1\ V_1^2$;
2) $V_1^1\ N_1^1\ V_2^2$;
3) $V_1^1\ N_2^1\ V_1^2$;
4) $V_1^1\ N_2^2\ V_2^2$;
5) $V_2^1\ N_1^1\ V_1^2$;
6) $V_2^1\ N_1^1\ V_2^2$;
7) $V_2^1\ N_2^1\ V_1^2$;
8) $V_2^1\ N_2^2\ V_2^2$;

Determinarea strategiei optime se face astfel: se determină varianta optimă în condiţii de risc sau incertitudine, după cum se cunosc sau nu probabilităţile $p(N_1^2)$ şi $p(N_2^2)$, pentru fiecare din nodurile decizionale $D_2 - D_5$, ceea ce echivalează cu rezolvarea problemelor decizionale din etapa a II-a; utilitatea care se obţine pentru fiecare din cele patru variante „optime" astfel găsite se va considera ca utilitate a consecinţei asociate modului decizional corespunzător. Va urma apoi aplicarea regulilor de decizie în condiţii de risc — dacă se cunosc $p(N_1^1)$ şi $p(N_2^1)$ — sau în condiţii de incertitudine — dacă cele două probabilităţi sînt necunoscute.

5. *Decizii de grup.* În sistemul axiomatic a lui von Neumann-Morgenstern, ca şi în alte sisteme de fundamentare a operării cu utilităţi, se consideră că, la nivelul unui decident individual, o alegere între două variante *a* şi *b* este raţională cînd se poate exprima precis că *a* este preferabil, echivalent sau nonpreferabil lui *b* şi dacă se respectă regula de tranzitivitate.

La nivelul unui grup de decizie, cerinţele de raţionalitate sînt mai complexe. J.K. Arrow defineşte cinci astfel de condiţii:

- *Condiţia 1.* Metoda de decizie colectivă trebuie să fie aplicabilă mulţimii tuturor variantelor posibile.
- *Condiţia 2.* Dacă o anumită variantă urcă pe scara preferinţelor fiecărui individ, atunci ea trebuie să urce pe scara preferinţelor grupului i.
- *Condiţia 3.* Dacă decizia se referă la *n* alternative posibile, „clasamentul" făcut de un grup acestora nu trebuie să fie modificat prin luarea în considerare a unei noi variante. De exemplu, dacă se compară variantele *a* şi *b*, prima fiind preferată şi se ia în considerare varianta *c*, relaţia între *a* şi *b* nu trebuie să se modifice.
- *Condiţia 4.* Regula după care se extrage decizia colectivă nu trebuie să fie independentă de opiniile individuale, ci trebuie să depindă direct de acestea.
- *Condiţia 5.* Decizia colectivă nu trebuie să fie identică cu opinia unui anumit membru al grupului, fără a ţine seama de opiniile celorlalţi.

Plecînd de la aceste condiţii Arrow demonstrează că nu există nici o metodă de decizie colectivă care să satisfacă cele cinci condiţii de raţionalitate enunţate şi care să ducă întotdeauna la o soluţie corectă cînd numărul decidenţilor este mai mare sau egal cu 2, iar numărul alternativelor superior lui 2. Acest rezultat, denumit *paradoxul lui Arrow*, a fost mult dezbătut şi analizat, propunîndu-se o serie de metode pentru ieşirea din impas.

Una dintre metode constă în renunţarea la condiţia a 3-a a lui Arrow, ceea ce echivalează cu posibilitatea de a „grada", preferinţele şi, în ultimă

Scanned with OKEN Scanner

instanță, cu introducerea unor utilități aditive interpersonale. Aditivitatea interpersonală a utilității este însă contestată de numeroși autori.

O altă cale constă în aplicarea *metodei ELECTRE* (paragraful 8.3).

Deciziile de grup sînt procese de mare frecvență și importanță în munca de organizare și conducere a activității în întreprinderi, unde cea mai mare parte a hotărîrilor se iau în colective de specialiști. Acest fapt justifică pe deplin analiza teoretică a fenomenului și încercarea de modelare matematică, fie și parțială deocamdată, pe care o întreprinde teoria deciziilor de grup.

8.3. Clasament și alegere după criterii multiple

În anul 1965 un grup de cercetători de la SEMA * au pus bazele binecunoscutei metode de *„clasament și alegere în prezența unor puncte de vedere multiple"*, cunoscută sub numele de *metoda ELECTRE*, aplicabilă în rezolvarea a numeroase probleme de decizie multidimensionale.

Această metodă permite compararea și clasificarea elementelor unei mulțimi de obiecte $\mathbf{M}$, ținînd seama de m puncte de vedere, în scopul găsirii unor submulțimi omogene de elemente, precum și pentru a stabili o ierarhizare a elementelor.

Fie o mulțime $\mathbf{M} = \{\mathbf{M}_1, \ldots, \mathbf{M}_m\}$, ale cărei elemente le vom numi strategii sau variante și n criterii sau puncte de vedere $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ pe baza cărora judecăm cele m obiecte $\mathbf{M}_1, \ldots, \mathbf{M}_m$. Criteriile după care sînt, judecate obiectele mulțimii $\mathbf{M}$ pot avea importanțe și semnificații diferite stabilindu-se pentru fiecare problemă în parte :

— prezența sau absența unei proprietăți ;

— recunoașterea unei caracteristici sau aprecieri a unui factor calitativ ;

— atribuirea unei note sau evaluarea unui factor cantitativ.

Analiza unui element $\mathbf{M}_i$ din punctul de vedere $\mathbf{X}_j$ duce la un rezultat a_{ij}, care poate fi o valoare numerică, o notă atribuită elementului $\mathbf{M}_i$, sau o apreciere calitativă (rău, bun, foarte bun etc.).

Fiecărui element a_{ij}, i se poate asocia o notă de apreciere care poate fi „normalizată" la o anumită scară. Elementele și criteriile, împreună cu rezultatele, se transpun într-un tablou ca în cazul de mai jos (tabelul 8.11) :

Tabelul 8.11

	$\mathbf{X}_1$	$\mathbf{X}_2$	...	$\mathbf{X}_n$
$\mathbf{M}_1$	a_{11}	a_{12}	...	a_{1n}
$\mathbf{M}_2$	a_{21}	a_{22}	...	a_{2n}
.	.	.		.
.	.	.		.
.	.	.		.
$\mathbf{M}_m$	a_{m1}	a_{m2}	...	a_{mn}

* Société d'Economie et Mathematique appliquée, Paris.

Scanned with OKEN Scanner

Dacă elementele mulțimii **M** sînt modurile posibile de a acționa ale unui decident, iar elementele mulțimii **X** — criteriile de decizie, sîntem în fața unei probleme tipice de decizie multidimensională.

Metoda ELECTRE permite ordonarea alternativelor după criterii complexe și deci alegerea celei mai bune alternative, ținînd seama atît de aspectele favorabile, cît și de cele nefavorabile ale diferitelor posibilități.

Se notează, pentru fiecare criteriu, cu $\overline{\mathbf{X}}_j = (a_{1j}, a_{2j}, \ldots, a_{mj})$ mulțimea rezultatelor pe care le vom numi stări, la care poate conduce analiza din punctul de vedere $\mathbf{X}_j$ al elementelor lui **M**. În felul acesta se pot defini n aplicații care reflectă această asociere :

$$\gamma_j : \mathbf{M} \to \mathbf{X}_j \quad j = 1, \ldots, n.$$

Comparația elementelor mulțimii **M** după starea multidimensională a lui $\mathbf{X} = \overline{\mathbf{X}}_1 \times \overline{\mathbf{X}}_2 \times \ldots \times \overline{\mathbf{X}}_n$, care se asociază prin intermediul aplicațiilor γ_j, nu este posibilă decît dacă oricare două elemente ale lui **M** se pot compara din punctul de vedere al fiecărui criteriu. Acest lucru ne va conduce la a asocia fiecărui criteriu $\mathbf{X}_j$ un graf $\mathbf{G}_j = (\mathbf{M}, \mathbf{U}_j)$, unde :

a) elementele mulțimii **M** vor constitui vîrfurile grafului :

b) între două vîrfuri $\mathbf{M}_i$ și $\mathbf{M}_k$ vom duce un arc, orientat de la $\mathbf{M}_i$ spre $\mathbf{M}_k$ dacă $\mathbf{M}_k$ depășește pe $\mathbf{M}_i$ în ordonarea după criteriul $\mathbf{X}_j$, dacă în această ordonare $\mathbf{M}_i$ și $\mathbf{M}_k$ sînt echivalente se va trasa între ele un arc în ambele sensuri.

Graful $\mathbf{G}_i$ ne dă o reprezentare clară a relației de preordine definită pe **M** de criteriul $\mathbf{X}_j$. Acest graf are următoarele proprietăți :

a) este tranzitiv, adică, dacă între două vîrfuri oarecare $\mathbf{M}_i$ și $\mathbf{M}_k$ există un drum orientat de la $\mathbf{M}_i$ spre $\mathbf{M}_k$, atunci există un arc și de la $\mathbf{M}_i$ la $\mathbf{M}_k$;

b) este complet, adică oricare două vîrfuri $\mathbf{M}_i$ și $\mathbf{M}_k$ sînt unite printr-un arc cel puțin într-un sens.

Dispunînd deci de posibilitatea de a ordona elementele mulțimii **M** după fiecare criteriu în parte, trebuie găsit un procedeu de a „sintetiza" rezultatul acestor comparații într-o ordonare finală. Acest lucru revine la obținerea unui graf-sinteză $\mathbf{G}_0 = (\mathbf{M}, \mathbf{U}_0)$ a celor n grafe $\mathbf{G}_j$, de la care să se poată deduce o ordonare a elementelor lui **M**, ținînd seama de toate criteriile $\mathbf{X}_j$.

În graful $\mathbf{G}_0$, prin definiție, avem arc de la un vîrf $\mathbf{M}_k$ la un vîrf $\mathbf{M}_i$ dacă și numai dacă există arc de la $\mathbf{M}_k$ la $\mathbf{M}_i$ în toate grafele $\mathbf{G}_j (j = 1, \ldots, n)$.

Pentru a putea efectua o ordonare completă cu ajutorul grafului sinteză este necesar să se introducă conceptele de concordanță și discordanță care vor permite în final completarea lui $\mathbf{G}_0$ cu arce pînă se obține o soluție „bună" din toate punctele de vedere.

Indicatorul de concordanță între două strategii $\mathbf{M}_g$ și $\mathbf{M}_h$ se definește astfel :

$$c(\mathbf{M}_g, \mathbf{M}_h) = \frac{1}{k_1 + k_2 + \ldots + k_m} \sum_j \cdot k_j$$

Scanned with OKEN Scanner

unde indicii k_i $(i = 1, \ldots, n)$ sînt coeficienţii de importanţă ai criteriilor considerate, iar suma se face după acei indici j pentru care $a_{gj} \geqslant a_{hj}$. Din modul cum a fost definit, rezultă că acest indicator ia valori cuprinse între *0* şi *1*. El va da o indicaţie asupra nivelului de „depăşire" a unei strategii $\mathbf{M}_g$ de către o altă strategie $\mathbf{M}_h$.

Indicatorul de discordanţă între două strategii $\mathbf{M}_g$ şi $\mathbf{M}_h$ se defineşte astfel:

$d(\mathbf{M}_g, \mathbf{M}_h) = 0$ dacă $a_{hj} < a_{gj}$ oricare ar fi $j = 1,2,\ldots, n$;

$$d(\mathbf{M}_g, \mathbf{M}_h) = \frac{1}{d} \max \mid a_{gj} - a_{hj} \mid,$$ maximul luîndu-se pentru acei indici j

pentru care $a_{hj} \geqslant a_{gj}$, iar d reprezintă ecartul maxim între valorile stărilor[1].

Dacă indicatorul de concordanţă dă indicaţii despre modul în care o strategie $\mathbf{M}_h$ „depăşeşte" (din punctul de vedere luat în considerare) strategia $\mathbf{M}_g$, indicatorul de discordanţă ne dă o informaţie similară (dacă nu identică) despre modul în care strategia $\mathbf{M}_h$ este depăşită de strategia $\mathbf{M}_g$.

În mulţimea $\mathbf{M}$ se introduce o relaţie de surclasare definită astfel: o strategie $\mathbf{M}_i$ surclasează o strategie $\mathbf{M}_k$ dacă $c(\mathbf{M}_i, \mathbf{M}_k) \geqslant p$ şi $d(\mathbf{M}_i, \mathbf{M}_k) \leqslant q$, unde q şi p sînt două valori-prag, alese de decident şi cuprinse între *0* şi *1*, valoarea lui p fiind apropiată de *1*, iar cea a lui q — relativ apropiată de *0*.

Acestei relaţii de surclasare, pentru fiecare pereche de valori p şi q, i se asociază un graf:

$$\mathbf{G}(p, q) = (\mathbf{M}, \mathbf{U}(p, q))$$

în care:

$$[(\mathbf{M}_i, \mathbf{M}_k) \in \mathbf{U}(p, q)] \Leftrightarrow [c(\mathbf{M}_i, \mathbf{M}_k) \geqslant p \text{ şi } d(\mathbf{M}_i, \mathbf{M}_k) \leqslant q].$$

Din definiţia grafului $\mathbf{G}(p, q)$ rezultă că dacă $p \leqslant p'$ şi $q \geqslant q'$, atunci $\mathbf{G}(p', q')$ este un subgraf parţial al lui $\mathbf{G}(p, q)$. Cu alte cuvinte, micşorînd pragul de concordanţă p, şi mărind pragul de discordanţă q, se introduc în graful $\mathbf{G}(p, q)$ legături noi.

În practică, se porneşte de la valorile $p = 1$ şi $q = 0$ şi se micşorează valoarea lui p, iar cea a lui q se măreşte, pînă cînd se obţine o variantă care le surclasează pe toate celelalte.

O anumită variantă $\mathbf{M}^*$ a mulţimii $\mathbf{M}$ le va surclasa, pe toate celelalte cînd $c(\mathbf{M}^*, \mathbf{M}_i) \geqslant p$ şi $d(\mathbf{M}^*, \mathbf{M}_i) \leqslant q$, pentru $j = 1,2,\ldots, n$, surclasarea fiind cu atît mai puternică cu cît p este mai apropiat de *1*, iar q mai apropiat de *0*.

Dată fiind frecventa utilizare a metodei ELECTRE în cele mai diverse probleme de decizie multidimensională, să ilustrăm procedeul respectiv printr-un exemplu.

Înainte de a trece la calcule vom arăta că procedeul de atribuire a „notelor" pe care îl propune metoda este insuficient precizat, ceea ce poate

---

[1] Dacă elementele a_{ij} pentru un anumit criteriu sînt calitative, se vor înlocui în mod obligatoriu cu note.

Scanned with OKEN Scanner

constitui sursa unor confuzii la calculul indicatorilor de discordanță. Într-adevăr, din expunerea metodei pare că în aceeași problemă este posibilă coexistența unor criterii cantitative, pentru care valorile a_{ij} au semnificații fizice, economice etc., alături de criterii calitative pentru care valorile a_{ij} sînt simple note de apreciere. Apare o neclaritate asupra semnificației conceptului de discordanță, întrucît diferențele $a_{gj} - a_{kj}$ din expresia lui $d(\mathbf{M}_g, \mathbf{M}_h)$ nu mai sînt comparabile. Acest neajuns este, în fond, consecința faptului că metoda ELECTRE ignoră complet folosirea utilităților. Nu ne referim neapărat la estimarea unor utilități prin metode care sînt uneori complicate, ci, pur și simplu, la introducerea unei unități comune de estimare pentru toate elementele a_{ij} fie ele cantitative sau calitative.

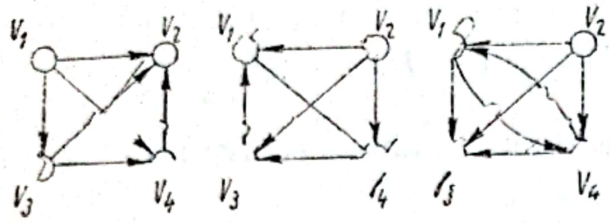


Fig. 8.2.

Pentru a înlătura această neclaritate ne-am propus să interpretăm că în fond intențiile autorilor metodei au fost acelea de a atribui rezultatelor a_{ij}, semnificația unor utilități, estimate printr-o metodă oarecare. Așadar, am transcris în tabelul 8.12 utilitățile consecințelor unei probleme decizionale cu patru variante și trei criterii decizionale după care am trecut în continuare la construirea grafelor $(\mathbf{M}, \mathbf{U}_i)$ (fig. 8.2).

Tabelul 8.12

Variante	Utilități		
	Pentru criteriul $\mathbf{C}_1$	Pentru criteriul $\mathbf{C}_2$	Pentru criteriul $\mathbf{C}_3$
$\mathbf{V}_1$	1	0	0,8
$\mathbf{V}_2$	0	1	1
$\mathbf{V}_3$	0,7	0,2	0
$\mathbf{V}_4$	0,4	0,8	0,8

Considerînd că coeficienții de importanță ai celor trei criterii ale problemei au fost estimați precum urmează, $k_1 = 2$, $k_2 = 2$; $k_3 = 1$, putem trece la calculul indicatorilor de concordanță:

$$c(\mathbf{V}_1, \mathbf{V}_2) = \frac{1}{5}(2) = 0,4 \qquad c(\mathbf{V}_1, \mathbf{V}_3) = \frac{1}{6}(2+1) = 0,6$$

$$c(\mathbf{V}_1, \mathbf{V}_4) = \frac{1}{5}(2+1) = 0,6 \qquad c(\mathbf{V}_2, \mathbf{V}_1) = \frac{1}{5}(2+1) = 0,6$$

$$c(\mathbf{V}_2, \mathbf{V}_3) = \frac{1}{5}(2+1) = 0,6 \qquad c(\mathbf{V}_2, \mathbf{V}_4) = \frac{1}{5}(2-1) = 0,6$$

Scanned with OKEN Scanner

$$c(V_3, V_1) = \frac{1}{5}(2) = 0,4 \qquad c(V_3, V_2) = \frac{1}{5}(2) = 0,4$$

$$c(V_3, V_4) = \frac{1}{5}(2) = 0,4 \qquad c(V_4, V_1) = \frac{1}{5}(2+1) = 0,6$$

$$c(V_4, V_2) = \frac{1}{5}(2) = 0,4 \qquad c(V_4, V_3) = \frac{1}{5}(2+1) = 0,6$$

care se înscriu în tabelul 8.13.

Tabelul 8.13

V_i \ V_j	V_1	V_2	V_3	V_4
V_1		0,4	0,6	0,6
V_2	0,6		0,6	0,6
V_3	0,4	0,4		0,4
V_4	0,6	0,4	0,6	

În continuare calculăm indicatorii de discordanţă:

$$d(V_1, V_2) = \frac{1}{1} \cdot 1 = 1 \qquad d(V_1, V_3) = \frac{1}{1} \cdot 0,2 = 0,2$$

$$d(V_1, V_4) = \frac{1}{1} \cdot 0,8 = 0,8 \qquad d(V_2, V_1) = \frac{1}{1} \cdot 1 = 1$$

$$d(V_2, V_3) = \frac{1}{1} \cdot 0,7 = 0,7 \qquad d(V_2, V_4) = \frac{1}{1} \cdot 0,4 = 0,4$$

$$d(V_3, V_1) = \frac{1}{1} \cdot 0,8 = 0,8 \qquad d(V_3, V_2) = \frac{1}{1} \cdot 0,8 = 0,8$$

$$d(V_3, V_4) = \frac{1}{1} \cdot 0,8 = 0,8 \qquad d(V_4, V_1) = \frac{1}{1} \cdot 0,6 = 0,6$$

$$d(V_4, V_2) = \frac{1}{1} \cdot 0,2 = 0,2 \qquad d(V_4, V_3) = \frac{1}{1} \cdot 0,3 = 0,3$$

înscriindu-i în tabelul 8.14.

Cea mai puternică surclasare va fi între variantele V_2 şi V_1, pentru care avem $p(V_1, V_3)=0,6$ şi $q(V_1, V_3)=0,2$, conform figurii 8.3, *a*. Dacă scădem

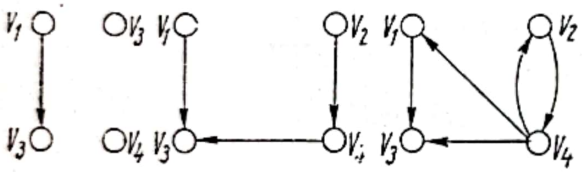
Scanned with OKEN Scanner

„nivelul" surclasării la $p = 0,6$; $q = 0,4$ apar 3 relații de surclasare (fig. 8.3, *b*) și de-abia la $p = 0,4$; $q = 0,6$ (fig. 8.3,*c*) alternativa $\mathbf{V}_4$ le va surclasa pe toate celelalte putînd fi considerată „optimă" (după această metodă).

Tabelul 8.14

V_i \ V_k	V_1	V_2	V_3	V_4
V_1		1	0,5	0,8
V_2	1		0,7	0,4
V_3	0,8	0,8		0,8
V_4	0,6	0,2	0,3	

O observație pe care o vom trece la activul metodei este simplitatea ei (în special în comparație cu metodele de estimare a utilității și de testare a independenței criteriilor propuse de școala americană). Tot la activ vom trece luarea în considerare a discordanței între consecințe, ceea ce reprezintă un element nou față de metodele clasice de comparare a utilității.



a. b. c.

Fig. 8.3.

Revenind însă la observațiile critice, vom arăta că cei doi coeficienți pe care îi definește metoda au semnificații deosebite: $c(\mathbf{M}_g, \mathbf{M}_h)$ este un raport de sume a unor coeficienți de importanță, iar $d(\mathbf{M}_g, \mathbf{M}_h)$ este un raport de ecarturi între note. De asemenea, utilizarea coeficienților de importanță mărește gradul de subiectivism al procedeului.

Considerăm că procesul de comparare și clasificare ar fi mai bine de realizat dacă cei doi coeficienți ar avea aceeași dimensiune.

În acest sens am putea propune:

— coeficientul normalizat de concordanță;

$$c^*(\mathbf{M}_g, \mathbf{M}_h) = \sum_j k_j \, (a_{gj} - a_{hj})$$

în care suma se face după acei indici j pentru care $a_{gj} > a_{hj}$;

— coeficientul normalizat de discordanță:

$$d^*(\mathbf{M}_g, \mathbf{M}_h) = \sum_j k_j (a_{hj} - a_{gj})$$

în care suma se face după acei indici j pentru care $a_{hj} > a_{gj}$.

Scanned with OKEN Scanner

Pentru valorile-prag p și q în acest caz cuprinse între *0* și *1* se normalizează coeficienții de importanță a criteriilor astfel ca $\Sigma k_j = 1$, precum și valorile „notelor" astfel ca $0 \leqslant a_{ij}^z \leqslant 1$.

Cîteva aplicații efectuate cu acești doi coeficienți ne-au condus la concluzia că ei reflectă mai bine fenomenul de asemănare-deosebire. Acest lucru poate fi observat și din faptul că, $c^*(\mathbf{M}_g, \mathbf{M}_h) = d^*(\mathbf{M}_h, \mathbf{M}_g)$, adică „concordanța" între $\mathbf{M}_h$ și $\mathbf{M}_g$ este egală cu „discordanța" între $\mathbf{M}_h$ și $\mathbf{M}_g$, ceea ce pare firesc.

În fine o ultimă observație, și anume metoda ELECTRE nu ține seama în nici un fel de dependența, respectiv independența criteriilor.

Observațiile de mai sus nu scad întru nimic meritul școlii franceze * care, prin priceperea sa de a găsi structuri raționale care să permită compararea și clasificarea după mai multe criterii contribuie substanțial la închegarea teoriei deciziilor multidimensionale.

8.4. Programarea liniară cu mai multe funcții obiectiv

În paragraful 8.2 am arătat că problemele de programare liniară sînt probleme de decizie cu o infinitate de variante. O categorie aparte o reprezintă problemele de programare liniară cu mai multe funcții-obiectiv.

Avînd în vedere faptul că funcțiile obiectiv au semnificația unor criterii decizionale, am considerat oportună tratarea acestei probleme în cadrul prezentului capitol.

Problema programării liniare cu mai multe funcții obiectiv, denumită și programarea liniară multidimensională a apărut în literatura de specialitate în ultimii 10—15 ani, negăsindu-și încă, pînă în prezent, o rezolvare definitivă.

Diverse metode propuse, pe care le vom expune în continuare reprezintă etape de lucru, încercări de a găsi soluții operaționale satisfăcătoare, fiecare dintre ele avînd, mai mult sau mai puțin, părți tari și puncte slabe.

Fie problema de programare liniară multidimensională

$$\mathbf{A}\mathbf{X}' \leqslant \mathbf{b} \tag{8.1}$$

$$\mathbf{X} \geqslant 0 \tag{8.2}$$

$$(\text{optimum})\ \mathbf{F} = \mathbf{C}\mathbf{X}' \tag{8.3}$$

unde :

$$\mathbf{A} = ((a_{ij})) \qquad \begin{matrix} i = 1, \ldots, m \\ j = 1, \ldots, n \end{matrix} \qquad \text{este o matrice } (m \times n)$$

* Lucrările privind problemele de decizie multidimensională elaborate de grupul condus de Prof. B. Roy de la Univ. din Paris.

Scanned with OKEN Scanner

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} \in \mathbf{R}^n \text{ este un vector coloană}$$

$\mathbf{X}' = (x_1, x_2, \ldots, x_n)$ este un vector coloană necunoscut

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \mathbf{F}_r \end{bmatrix} \text{ vector coloană cu } r \text{ componente care sînt funcţii de eficienţă}$$

$$\mathbf{C} = ((c_{ij})) \quad \begin{matrix} j = 1, \ldots, r \\ i = 1, \ldots, n \end{matrix} \quad \text{este o matrice } (r \times n).$$

Pentru fixarea ideilor şi fără a restrînge generalitatea vom presupune în cele ce urmează că toate funcţiile se cer a fi maximizate. Dacă nu este îndeplinită această condiţie, prin transformarea $\min \mathbf{F}(\mathbf{X}) = -\max[-\mathbf{F}(\mathbf{X})]$ putem face ca toate funcţiile să fie de maxim.

Notăm cu $\mathbf{D}$ mulţimea vectorilor $\mathbf{X}' = x_1, x_2, \ldots, x_n)$ care verifică sistemul (8.1) — (8.2) şi pe care îl vom numi domeniul soluţiilor admisibile.

Cele r funcţii-criteriu de eficienţă exprimă mărimi distincte şi nereductibile la o mărime comună (cînd două sau mai multe dintre funcţiile de eficienţă exprimă mărimi reductibile la o mărime comună, de pildă valoarea în bani, apare evident posibilitatea înlocuirii lor cu o funcţie-sumă).

Fiecare din cele r funcţii de eficienţă defineşte împreună cu sistemul (8.1) — (8.2) cîte o problemă de programare liniară a cărei rezolvare conduce la o soluţie optimă din punctul de vedere al criteriului considerat. Rezolvînd deci cele r probleme de programare liniară definite de sistemul (8.1) — (8.2) şi funcţiile (8.3), se obţin r soluţii optime, fiecare dintre acestea reprezentînd un vector $\mathbf{X}_h = (x_1^h, \ldots, x_n^h)$ ale cărui componente sînt acele valori ale variabilelor x_j care verifică sistemul (8.1) — (8.2) şi maximizează (sau minimizează) valorile funcţiilor-criteriu $\mathbf{F}_h$.

Problema de programare cu criterii multiple, din ce în ce mai prezentă în literatura de specialitate din ultimii ani, propune găsirea unui vector $\mathbf{X}^x = (x_1^x, \ldots, x_n^x)$ care verificînd sistemul (8.1) — (8.2) să fie „cît mai bun" din punctul de vedere al ansamblului funcţiilor-criteriu de eficienţă $\mathbf{F}_h$, $h = 1, \ldots, r$.

Obiectivul urmărit nu este de loc clar întrucît, pînă în prezent cel puţin, nu există o accepţiune precisă şi unanim admisă asupra unei mărimi care să fie cît mai bună din mai multe puncte de vedere.

În acest sens, apare evidentă legătura între problema de programare liniară cu criterii multiple şi problema deciziilor multidimensionale.

O consecinţă a acestei stări de lucruri este apariţia mai multor metode pentru rezolvarea problemei de programare liniară cu criterii multiple, fiecare corespunzînd unui anumit punct de vedere asupra optimumului multidimensional.

Scanned with OKEN Scanner

În continuare vom prezenta o serie de procedee recente în rezolvarea problemelor de programare liniară multidimensională.

• *Metoda optimizării utilităţilor globale* a fost elaborată de cercetători români [11] pornind de la faptul că problema programării liniare cu criterii multiple se poate reduce, în sens decizional, la problema deciziilor multidimensionale * . Ideea de bază a metodei este să se înlocuiască funcţiile de optim cu semnificaţie economică concretă prin funcţii de utilitate în sens von Neumann-Morgenstern care vor putea fi însumate pentru a obţine funcţia-sinteză dorită.

Se începe prin a determina, pentru fiecare din cele r funcţii de eficienţă, cele două valori extreme ** una dintre ele fiind valoarea optimă, iar cealaltă, valoarea cea mai dezavantajoasă pe care o vom numi *valoarea pessimă* *** pes $\mathbf{F} = -\text{opt}\ (-\mathbf{F})$.

Se rezolvă deci r probleme de programare liniară determinate de sistemul (8.1) — (8.2) şi de funcţiile (8.3) obţinîndu-se valorile optime ale funcţiilor de eficienţă :

$$\mathbf{X}_1,\ \mathbf{X}_2, \ldots,\ \mathbf{X}_r;$$

similar se rezolvă cele r probleme de programare liniară determinate de (8.1) — (8.2) şi de funcţiile :

$$\text{pes}\ \mathbf{F} = -\text{opt}\ (-\mathbf{F}) \tag{8.4}$$

şi se găsesc r valori pessime : $\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_r$.

Pentru a transforma funcţiile (8.3) care au semnificaţii economice distincte în funcţii de utilitate, se rezolvă r sisteme de tipul :

$$\begin{aligned} \alpha_h \mathbf{X}_h + \beta_h &= 1 \\ \alpha_h \mathbf{Y}_h + \beta_h &= 0 \end{aligned}, \quad h = 1,2,\ldots,r \tag{8.6}$$

obţinîndu-se valorile α_h şi β_h.

Cu ajutorul coeficienţilor α_h şi β_h ($h = 1,2,\ldots, r$) se transformă funcţiile (8.3) în funcţii de utilitate, făcînd transformarea liniară următoare :

$$\mathbf{F}'_h = \sum_{j=1}^{n} \alpha_h c_{jh} x_j + \beta_h \quad h = 1, 2, \ldots, r. \tag{8.7}$$

Pentru a determina funcţia-sinteză de utilitate $\mathbf{F}^*$ se stabilesc coeficienţii de importanţă $\pi_h (h = 1, 2, \ldots, r)$ pentru fiecare funcţie-obiectiv iniţială, după care se scrie :

$$\mathbf{F}^* = \sum_{h=1}^{r} \pi_h \mathbf{F}'_h = \sum_{h=1}^{r} \sum_{j=1}^{1} \pi_h (\alpha_h c_{jh} x_j + \beta_h).$$

---

* A se vedea paragraful 8.2.

** Am presupus, pentru simplificare, că ambele valori extreme sînt finite. Cînd una dintre ele este la infinit, va trebui adăugată o restricţie suplimentară care să transforme problema într-una cu ambele valori extreme finite.

*** Termenul „pessim" se va interpreta ca opusul lui optim.

Scanned with OKEN Scanner

Se înţelege că la adunare se va ţine seama de semnificaţia operaţiei de optimizare pentru fiecare funcţie $\mathbf{F}_h$ acordindu-se semnul „+" pentru maximizare şi semnul „—" pentru minimizare, putîndu-se considera, în felul acesta, că funcţia F urmează a fi maximizată.

Rezolvarea problemei de programare liniară definită de sistemul de relaţii (8.1) — (8.2) şi de funcţia de optim.

$$\text{opt } \mathbf{F}^* = \sum_{h=1}^{r_1} k_h \mathbf{F}'_h$$

va conduce la o soluţie de maximă utilitate, care ţine seama de toate funcţiile de optim ale problemei.

În ultimii ani şcoala franceză a elaborat o serie de lucrări în care propune soluţii ingenioase pentru problema programării liniare multidimensionale.

• Una dintre aceste soluţii este dată de *metoda POP* elaborată de R. Benayoun şi J. Tergny * în care se urmăreşte obţinerea unui vector X care realizează „cel mai bun compromis".

În mulţimea soluţiilor admisibile **D** se defineşte următoarea relaţie de ordine parţială ** .

Fiind date două soluţii admisibile $\mathbf{X}_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ şi $\mathbf{X}_2(x_{21}, x_{22}, \ldots, x_{2n})$, spunem că soluţia $\mathbf{X}_1$ este preferabilă soluţiei $\mathbf{X}_2$ (sau că $\mathbf{X}_1$ domină uniform pe $\mathbf{X}_2$) şi vom nota $\mathbf{X}_1 \geqslant \mathbf{X}_2$ dacă :

$$\mathbf{F}_j(\mathbf{X}_1) = \sum_{i=1}^{n} c_{ji} x_{1i} \geqslant \mathbf{F}_j(\mathbf{X}_2) = \sum_{i=1}^{n} c_{ji} x_{2i};$$

$$j = 1, 2, \ldots, j_0 - 1, j_0 + 1, \ldots, r$$

şi

$$\mathbf{F}_{j_0}(\mathbf{X}_1) = \sum_{i=1}^{n} c_{j_0 i}\ \mathbf{X}_{i1} > \mathbf{F}_{j_0}(\mathbf{X}_2) = \sum_{i=1}^{n} c_{j_0 i} x_{2i}.$$

Se numeşte soluţie eficace orice soluţie maximală în sensul ordinii definite mai sus, adică o soluţie **X** cu proprietatea că oricare ar fi soluţia **Y**, astfel încît $\mathbf{Y} \geqslant \mathbf{X}$, rezultă $\mathbf{X} \sim \mathbf{Y}$ (soluţia **X** este egal preferată cu **Y**).

Vectorul $\mathbf{S}_j = (x_{j1}, x_{j2}, \ldots, x_{jn})'$ se numeşte soluţie marginală relativ la criteriul $\mathbf{F}_j$, dacă este o soluţie a sistemului (8.1) — (8.2) şi maximizează funcţia $\mathbf{F}_j(\mathbf{X}) = \sum_{i=1}^{n} c_{ji} x_i$.

---

* Benaioun R., Tergny J., Critères multiples en programmation mathématique, în „Revue Française d'Informatique et de Recherche Operationnelle, nr. V-2, 1969.

** S-au preluat notaţiile originale din articolul citat mai sus.

Scanned with OKEN Scanner

Pentru fiecare din cele r funcţii de eficienţă $\mathbf{F}_j$ se poate găsi cîte o soluţie marginală $\mathbf{S}_j$ şi de aici se poate construi tabelul consecinţelor avînd r linii şi r coloane, (tabelul 8.15).

Tabelul 8.15

Criterii / Soluţii marginale	$\mathbf{F}_1$	$\mathbf{F}_2$	...	$\mathbf{F}_r$
$\mathbf{S}_1$	y_{11}	y_{12}	...	y_{1r}
$\mathbf{S}_2$	y_{21}	y_{22}	...	y_{2r}
.	.	.		.
.	.	.		.
.	.	.		.
$\mathbf{S}_r$	y_{r1}	y_{r2}	...	y_{rr}

unde $y_{ij} = \mathbf{F}_j(\mathbf{S}_i) = \sum_{k=1}^{n} c_{jk} x_{ik}$.

Se numeşte soluţie mixtă asociată soluţiilor realizabile $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_p$ orice soluţie de forma :

$$\mathbf{X} = \sum_{k=1}^{p} \lambda_k \mathbf{S}_k ; \sum_{k=1}^{p} \lambda_k = 1, \ 0 \leqslant \lambda_k \leqslant 1 \quad 1 \leqslant k \leqslant p.$$

Soluţiile mixte asociate la soluţiile $\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_p$ sînt puncte (diferite de vîrfuri) ale celui mai mic poliedru convex $\mathbf{P} = \mathbf{P}(\mathbf{S}_1, \ldots, \mathbf{S}_p)$ care conţine punctele $\mathbf{S}_1, \ldots, \mathbf{S}_p$.

Evident, dacă soluţiile $\mathbf{S}_1, \ldots, \mathbf{S}_p$ sînt realizabile atunci şi soluţiile mixte asociate sînt realizabile.

Dacă $\mathbf{X} = \sum_{k=1}^{r} \lambda_k \mathbf{S}_k$ este o soluţie mixtă, atunci din tabelul consecintelor se pot deduce valorile sale relativ la fiecare criteriu :

$$\mathbf{F}_j(\mathbf{X}) = \sum_{k=1}^{r} \lambda_k \mathbf{Y}_{kj} \quad j = 1, \ldots, r.$$

În tabelul consecinţelor se poate presupune că toate soluţiile marginale sînt eficace. Dacă, spre exemplu, soluţia marginală $\mathbf{S}_j$, nu este eficace, aceasta se înlocuieşte cu soluţia obţinută prin rezolvarea problemei :

$$\max \mathbf{F}_{j_0} = \mathbf{F}_{j_0}(\mathbf{X}) + \varepsilon \Sigma \mathbf{F}_j(\mathbf{X}) \quad (\varepsilon > 0 \text{ suficient de mic})$$

$$\mathbf{AX} \leqslant \mathbf{b}$$

$$\mathbf{X} \geqslant \mathbf{O}$$

care se demonstrează că este eficace.

Scanned with OKEN Scanner

Metoda POP este o procedură de explorare secvențială a soluțiilor eficace, decidentul avînd posibilitatea de a interveni pe parcursul rezolvării problemei și a diviza procesul de căutare pentru a găsi soluții convenabile cît mai eficiente.

Metoda este iterativă, fiecare iterație descompunîndu-se în două etape, care sînt :

Faza A, se pornește de la tabelul consecințelor asociat soluțiilor eficace $S_1, S_2, \ldots, S_r$ și se consideră poliedrul $P = P(S_1, S_2, \ldots, S_r)$ generat de aceste puncte.

Pentru a reduce numărul soluțiilor eficace luate în considerare se pot imagina diverse metode de restrîngere a domeniului D al soluțiilor admisibile și anume prin eliminarea unor vîrfuri S_j și prin introducerea altora (în număr mic) se construiește șirul de poliedre P_j astfel încît :

$$P \supset P_1 \supset \ldots \supset P_j \supset P_{j+1} \supset \ldots P_s = P'.$$

Decidentul se oprește la poliedrul P_s și trece la faza B atunci cînd nu dorește să mai elimine sau să mai introducă noi vîrfuri sau cînd P_s se reduce la un singur punct s. În acest ultim caz dacă S este o soluție eficace, aceasta se consideră ca fiind soluția căutată dacă nu, se trece la faza B pentru a obține una mai bună. Dacă la sfîrșitul fazei A se găsește o soluție care, din punctul de vedere al decidentului, să dea satisfacție tuturor funcțiilor de eficiență, algoritmul se oprește.

Unul din procedeele de trecere de la poliedrul P_j la poliedrul P_{j+1}, numit de B. Benayonn și J. Tergny procedeul de comparare cu media, constă în adăugarea la soluțiile marginale a soluției medii S.

$$S = \frac{1}{p} \sum_{i=1}^{p} S_i$$

și apoi compararea fiecărui vîrf S_i cu media S pentru a decide dacă poate fi eliminată sau nu.

Pentru aceasta se calculează două valori :

p_j^* = proporția criteriilor F_j pentru care

$$y_j = \frac{1}{p} \sum_{i=1}^{p} y_{ij}$$

și

$$q_j^* = \max \left(- \min_j \frac{y_{ij} - \frac{1}{p} \sum_{i=1}^{p} y_{ij}}{M_j - m_j} ; 0 \right)$$

cu

$$M_j = \max\ y_{ij} \text{ și } m_j = \min\ y_{ij}$$

care se compară cu două valori-prag P și Q, eliminîndu-se soluțiile pentru care $p_j < P$ și $q_j > Q$.

Scanned with OKEN Scanner

Faza B. În această fază, decidentul face o reoptimizare pe un domeniu de admisibilitate restrîns, obţinut adăugînd sistemului de restricţii, altele, constînd din impunerea anumitor valori l_j^k unora din funcţiile obiectiv. Acest lucru revine la a distinge în domeniul $\mathbf{D}$ o submulţime $\mathbf{D}^k$ (la iteraţia k) care poate să conţină noi soluţii eficace. Domeniul $\mathbf{D}^k$ este deci intersecţia lui $\mathbf{D}$ cu soluţiile sistemului de inegalităţi $\mathbf{F}_j(\mathbf{X}) \geqslant l_j^k$, $j = 1, \ldots, m$.

Fie p^k şi p'^k poliedrele soluţiilor obţinute la etapa k. Valorile impuse unora dintre funcţiile obiectiv depind de valorile diferitelor vîrfuri ale poliedrelor p^k şi p'^k. Dacă

$$y_j' = \min_{\mathbf{X} \in p'^k} \mathbf{F}_j(\mathbf{X}) \quad j = 1, \ldots, m$$

şi

$$y_j^k = [\min_{\mathbf{X} \in p^k} \mathbf{F}_j(\mathbf{X})$$

se alege

$$l_j^k = y_j'^k.$$

Deoarece $y_j'^k \geqslant l_j^k$ înseamnă că orice soluţie din $\mathbf{D}^k$ va lua, pe toate criteriile, o valoare cel puţin egală cu cea mai mică din valorile luate de aceste criterii pe mulţimea punctelor care au fost eliminate.

Pe domeniul de admisibilitate $\mathbf{D}^k$ găsit, se calculează noile soluţii extreme luînd drept criterii vechile criterii şi un criteriu „mixt", sumă ponderată de criterii.

$$\mathbf{F}^k = \sum_j \pi_j^k \cdot \mathbf{F}_j(\mathbf{X}).$$

Un mod de a calcula coeficienţii π_j^k este următorul $\pi_j^k = 1/(y_j^k - \mathbf{F}_j(\bar{\mathbf{S}}'^k))$ unde $\bar{\mathbf{S}}'^k$ este soluţia medie pe p'^k.

La sfîrşitul acestei faze, decidentul poate reţine ca soluţie optimă una dintre soluţiile găsite, dacă aceasta prezintă interes sau poate continua cu faza A.

În spiritul metodei POP, cercetătorii francezi au elaborat o altă metodă, *STEM*, mai sistematică decît prima şi care uşurează aplicaţiile practice. Metoda STEM are avantajul că cere decidentului minimum de informaţie.

În această metodă, decidentul, printr-o alternare a fazei de calcul în cea de decizie, poate să dirijeze mai bine procesul de căutare a soluţiilor.

Metoda STEM constă în următoarele etape :

Etapa 1. Se rezolvă r probleme de programare liniară definite de sistemul (8.1) — (8.2) şi fiecare din funcţiile $\mathbf{F}_k$.

Fie :

$$\mathbf{Z}_1 = (\max_{\mathbf{X} \in \mathbf{D}} \mathbf{F}_1(\mathbf{X}), \ldots, \max \mathbf{F}_r(\mathbf{X})) = (\mathbf{F}_{11}, \mathbf{F}_{12}, \ldots, \mathbf{F}_{1r}),$$

vectorul ale cărui componente sînt valorile maxime ale funcţiilor de eficienţă.

Etapa 2. Se mai rezolvă încă o problemă de programare liniară definită de sistemul (8.1) — (8.2) şi o funcţie $\mathbf{F}^*$ sumă-ponderată de funcţiile

Scanned with OKEN Scanner

$F_1, \ldots, F_r$

$$F^* = \sum_{j=1}^{r} \pi_j F_j.$$

Stabilirea coeficienţilor π_j se face în funcţie de informaţia de care se dispune asupra importanţei relative a criteriilor.

Se consideră S valoarea maximă a funcţiei F şi vectorul :

$$Z(S) = (F_1(S), F_2(S), \ldots, F_r(S))$$

ale cărui componente sînt valorile funcţiilor de eficienţă $F_1, \ldots, F_r$ pentru soluţia S^*.

Etapa 3. Această etapă este prima în care intervine decidentul care este pus în situaţia de a răspunde dacă soluţia S^* conduce la valori acceptabile pentru funcţiile $F_1, \ldots, F_r$.

În acest scop, decidentul compară componentele vectorului $Z(S^*)$ cu cele ale vectorului Z_1.

Dacă toate funcţiile iau valori acceptabile pentru soluţia S^* problema este rezolvată. Dacă nu toate funcţiile iau valori acceptabile, decidentul trebuie să indice pentru funcţia F_k^x, care ia valoarea cea mai puţin satisfăcătoare, un prag F_k de la care valoarea sa îl va satisface.

Etapa 4. Se reia problema de la etapa 1, considerînd sistemul format de sistemul iniţial şi restricţia suplimentară introdusă în etapa 3.

Dacă $D_1 \subset D$ este noul domeniu al soluţiilor admisibile

$$AX \leqslant b$$

$$X \geqslant 0$$

$$F_k^x \geqslant F_k$$

etapa 1 ne va conduce la un nou vector :

$$Z_2 = (\max_{x \in D_1} F_1(X), \ldots, \max_{x \in D_1} F_r(X)) = (F_{21}, F_{22}, \ldots, F_{r2})$$

ale cărui componente sînt valorile maxime ale funcţiilor de eficienţă pe domeniul restrîns D_1.

Etapa 5. Este a doua etapă cu participarea decidentului. Acesta este solicitat să răspundă la întrebarea : „restrîngerea lui D duce, pentru unele funcţii, la valori maxime inadmisibile"?. Dacă se răspunde „da" se reia problema de la etapa 4 dar pentru un prag mai mic $F_k^{xx} < F_k^x$. Dacă răspunsul este „nu" problema se continuă cu etapa 2 pentru domeniul D_1.

În [11] se indică modul de alegere a coeficienţilor π_j folosiţi în etapa 2, în trei cazuri :

1. Se poate exprima importanţa fiecărei funcţii printr-un număr numit pondere.

2. Se cunosc relaţiile de importanţă între oricare două criterii.

3. Nu se dispune de nici o informaţie asupra importanţei funcţiilor F_h.

În primul caz, cunoscînd pentru fiecare criteriu F_j care este ponderea sa, p_j, se va lua $\pi_j = p_j$.

Scanned with OKEN Scanner

De remarcat este faptul că în întregul proces de calcul valorile π_j rămîn neschimbate.

Dacă se cunosc relaţiile de importanţă între criterii (al doilea caz), atunci oricare ar fi două criterii $\mathbf{F}_i$ şi $\mathbf{F}_j$, prin compararea acestora putem spune dacă sînt tot atît de importante, sau care este mai important.

Se asociază funcţiilor $\mathbf{F}_1, \ldots, \mathbf{F}_r$ o matrice pătratică.

$\mathbf{A} = ((a_{ij}))_{ij} = 1,2,\ldots, r$ ale cărei elemente a_{ij} se definesc astfel:

$$a_{ij} = \begin{cases} 1 \text{ dacă criteriile } \mathbf{F}_i \text{ şi } \mathbf{F}_j \text{ sînt de o importanţă egală;} \\ 2 \text{ dacă criteriul } \mathbf{F}_i \text{ este superior criteriului } \mathbf{F}_j; \\ 4 \text{ dacă criteriul } \mathbf{F}_i \text{ este cu mult superior lui } \mathbf{F}_j; \\ 0 \text{ în toate celelalte cazuri.} \end{cases}$$

Fie $p_i = \sum_{j=1}^{r} a_{ij}$. Valorile p_i ne dau o indicaţie asupra depăşirii de către criteriul $\mathbf{F}_i$ a celorlalte criterii. Se aleg coeficienţii π_i şi π_j astfel încît:

$$\frac{\pi_i}{p_i} = \frac{\pi_j}{p_j};\ i, j = 1, 2, \ldots, r.$$

Ca şi în cazul precedent, odată determinaţi coeficienţii π_i, aceştia rămîn neschimbaţi în diferite etape ale algoritmului.

Se consideră în continuare tabelul consecinţelor (tabelul 8.16). Fie $\mathbf{C}_s = (\mathbf{Z}_{1s}, \mathbf{Z}_{2s}, \ldots, \mathbf{Z}_{rs})$ o coloană a acestui tabel.

Tabelul 8.16

	$\mathbf{F}_1$	$\mathbf{F}_2$	...	$\mathbf{F}_j$	...	$\mathbf{F}_n$
$\mathbf{X}_1$						
$\mathbf{X}_2$						
.						
.						
.						
$\mathbf{X}_j$	$\mathbf{Z}_{j1}$	$\mathbf{Z}_{j2}$	...	$\mathbf{Z}_{jj}$	...	$\mathbf{Z}_{jn}$
.						
.						
.						
$\mathbf{X}_r$						

Se analizează „ecartul" între valorile pe coloană şi valoarea maximă pe acea coloană.

Apropierea valorilor pe coloană de valoarea maximă pe acea coloană indică un ecart de valori mic, ceea ce echivalează cu o importanţă mică a criteriului respectiv.

Invers, dacă ecartul de valori pentru un criteriu este mare, se acordă criteriului o pondere însemnată. Deoarece apropierea sau depărtarea valorilor faţă de valoarea maximă este relativă şi nu absolută, se trece de la tabelul 8.16 la un altul în care valorile Z_{ij} sînt înlocuite cu utilităţi.

Scanned with OKEN Scanner

Un criteriu $\mathbf{F}_s$ ia pentru valoarea maximă utilitatea egală cu *1* şi pentru valoarea minimă utilitatea egală cu *0*.

Fie m_k cea mai mică utilitate corespunzătoare funcţiei $\mathbf{F}_k$ şi fie $\alpha_k = 1 - m_k$.

Se aleg coeficienţii π_i şi π_j astfel încît :

$$\frac{\pi_i}{\alpha_i} = \frac{\pi_j}{\alpha_j}; \quad \sum_{i=1}^{r} \pi_i = 1.$$

Valorile coeficientului π_i, spre deosebire de primele două cazuri, se schimbă de la o iteraţie la alta a procesului de calcul.

* * *

Pentru a trage unele concluzii privind procedeele de rezolvare a problemei de programare multidimensionale vom arăta că metodele POP şi STEM au o poziţie specială prin caracterul lor de serioasă elaborare, ele reprezentînd o sinteză a numeroase procedee care le-au precedat. În acelaşi timp, însă, în ambele metode se găsesc idei discutabile, cum este aceea a folosirii coeficienţilor de pondere, a transformării în restricţii a soluţiilor optime găsite, precum şi ideea adăugării de restricţii noi problemei sau a efectuării altor modificări, faţă de datele iniţiale.

Metoda optimizării utilităţii globale, introducînd utilităţile în sensul definiţiei von Neumann-Morgenstern, permite incontestabil o mai bună însumare a funcţiilor de eficienţă păstrînd totodată o serie de slăbiciuni, derivate toate din caracterul subiectiv al utilităţilor, precum şi din necesitatea folosirii coeficienţilor de importanţă k_h. Vom mai menţiona şi caracterul prea automat al metodei optimizării utilităţii globale.

Considerăm că metodele POP şi STEM, precum şi optimizarea utilităţii globale au posibilităţi reale de aplicare, bineînţeles cu ajutorul calculatoarelor electronice şi că numai după un timp de utilizare practică se vor putea trage concluzii comparative privind eficienţa lor.

Scanned with OKEN Scanner

Capitolul IX

PROGRAMAREA OPERATIVĂ A PRODUCȚIEI

9.1. Aspecte generale

Programarea operativă a producţiei este una dintre cele mai complexe şi actuale probleme ale practicii economice la nivel de întreprindere, despre care se poate spune că este încă departe de a fi rezolvată ştiinţific şi eficient, chiar în ţările dezvoltate din punct de vedere industrial şi cu tradiţie în acest domeniu.

Una dintre caracteristicile cele mai tipice ale acestei probleme complexe este existenţa unei multitudini de înţelesuri care i se dau şi circulă în paralel, ca urmare tocmai a numărului mare de aspecte pe care le prezintă şi pe care adesea teoreticienii sau practicienii le privesc rupte din context, în funcţie de situaţia particulară cu care sînt confruntaţi.

În prezentul capitol vom încerca să trecem în revistă principalele probleme ale programării operative a producţiei, să arătăm cum se leagă acestea între ele şi să schiţăm unele idei privind posibilităţile de rezolvare teoretică şi practică.

Problemele programării operative a producţiei asupra cărora ne vom opri în continuare sînt următoarele :

— optimizarea structurii sortimentale a planului pe termen mediu şi scurt ;

— stabilirea necesarului de materii prime şi materiale corespunzător unui plan optim determinat ;

— unele aspecte tehnologice ale programării operative ;

— lansarea, urmărirea şi actualizarea producţiei ;

— unele aspecte ergonomice în programarea producţiei ;

— unele probleme psihosociologice ale programării operative. Să examinăm pe scurt caracteristicile fiecăreia dintre acestea.

Optimizarea structurii sortimentale a planului pe termen mediu (an, semestru, trimestru) şi pe termen scurt (lună, decadă, săptămînă, zi, oră) este o operaţie la frontiera dintre planificare şi programare operativă. Prin planul de termen mediu se stabilesc cantităţile care trebuie produse pînă la nivel pe trimestru, putîndu-se optimiza structura sortimentală pe baza luării în considerare a restricţiilor globale privind disponibilul de capacităţi de producţie pe perioade foarte scurte (lună, decadă, zi), fără a

Scanned with OKEN Scanner

detalia sortimentele pe subansamble, repere și operații și acceptîndu-se de regulă ipoteza ritmicității producției. Trecerea de la această eșalonare-cadru, utilă pentru a da o informație globală asupra viitoarei desfășurări a producției, la o eșalonare în care încep să se ia în considerare aspectele de detaliu ale producției, în special cele privind utilizarea în timp a resurselor umane și materiale, reprezintă în fond momentul trecerii de la nivelul planificării la nivelul programării operative propriu-zise. Întrucît însă structura sortimentală a planului pe termen mediu reprezintă o dată de intrare în operația programării operative (pe termen scurt), specialiștii acesteia din urmă sînt interesați să participe la elaborarea planului sortimental pe termen mediu.

O problemă pe care adesea practicienii o asociază programării operative a producției este *stabilirea resurselor necesare pentru realizarea unui plan prestabilit*, îndeosebi a necesarului de materii prime și materiale. Aceasta este strîns legată de asigurarea resurselor, care, la rîndul ei, condiționează posibilitățile realizării produselor și sortimentelor din plan. Reversul acestei probleme este determinarea cantităților din anumite sortimente care se pot produce cu cantitățile de resurse existente la un moment dat. Ambele probleme privesc desigur, în primul rînd, pe specialiștii în tehnologie și aprovizionare, dar puternica influență pe care o exercită ele asupra realizării efective a planului justifică includerea lor în sfera preocupărilor privind programarea operativă.

Există numeroase *aspecte tehnologice în programarea operativă a producției*. Astfel, în afara problemei stabilirii necesarului de resurse, menționată mai sus, al cărui aspect tehnologic constă în determinarea structurii unui produs (componența în subansamble și repere) și a consumurilor specifice, una din problemele tehnologice cu profunde implicații în programarea operativă este proiectarea succesiunii operațiilor care conduc la obținerea produselor dorite sau, cu un alt termen, proiectarea tehnologiilor de fabricație. De asemenea, după o serie de caracteristici principale, specialiștii clasifică procesele de fabricație în cîteva clase mari (producție de unicate, producție de serie mică, mijlocie sau mare, cu tehnologie de grup, organizare a producției pe linii, în bandă etc.), apartenența unui anumit proces productiv la una din aceste clase avînd o influență hotărîtoare asupra procedeelor de programare operativă ce trebuie utilizate.

Lansarea, urmărirea și actualizarea producției este, fără îndoială, problema centrală a programării operative. Este vorba de determinarea în detaliu a reperelor sau loturilor de repere ce urmează a se executa în perioada scurtă considerată și de stabilirea momentelor cînd intră în lucru și cînd se termină fiecare reper sau lot, deci de eșalonarea în timp a operațiilor care se efectuează asupra reperelor și loturilor cu precizarea repartizării resurselor (mașini, materiale, forțe de muncă) pe aceste operații, în timp.

Acest proces complex de conducere operativă este denumit în literatura de specialitate anglo-saxonă *scheduling*, iar în cea franceză—*ordonnancement*.

A dispune de o metodologie eficientă de lansare, urmărire și actualizare a producției echivalează desigur cu a rezolva problema programării operative; experiența ne arată însă că încercarea de a rupe această problemă din context, adică de a o separa de celelalte enumerate mai sus, conduce la soluționări de multe ori inaplicabile practic.

Scanned with OKEN Scanner

Aspecte ergonomice și psihosociologice, în mod obișnuit considerate ca distincte de cele ale programării operative, trebuie, după opinia noastră, incluse în preocupările privind programarea operativă întrucît, oricît de perfecționate ar fi procedeele acesteia, nu se pot ignora aspectele ei umane, în ultimă analiză determinante în obținerea rezultatelor scontate.

Avînd în vedere aspectele prezentate mai sus, se pot distinge două direcții mari de abordare a problemei programării operative a producției :

1) abordarea generală, avînd drept obiectiv cuprinderea și rezolvarea problemei în toată complexitatea ei ;

2) abordarea pragmatică, urmărind găsirea unor soluții eficiente, aplicabile, uneori cu sacrificarea cerințelor de generalitate și rigurozitate științifică.

Abordarea generală a problemei programării operative a producției este în prezent în domeniul cercetării teoretice. Excluzînd tipurile foarte simple de probleme pentru care există soluții complete, rezolvarea generală teoretică și în special cea practică a problemelor de complexitate mare și chiar medie aparține desigur viitorului. Acest lucru se datorește atît dificultății de a găsi modele care să reflecte satisfăcător problema complexă a programării operative, cît, mai ales, imposibilității de a rezolva în prezent aceste modele, chiar cu ajutorul calculatoarelor electronice cele mai puternice.

În ceea ce privește abordarea pragmatică a problemei, se poate spune că în funcție de caracteristicile tehnologice, de tipul procesului de producție și de obiectivul urmărit, există o foarte largă gamă de soluții posibile.

Este esențial ca practicienii să accepte ideea că în practică nu se poate vorbi în prezent despre „problema programării operative a producției", ci de o multitudine de astfel de probleme. De aceea, prima grijă a analistului care vine în contact cu o problemă din această categorie este să încerce definirea ei precisă, stabilirea subproblemelor și încadrarea lor în grupele menționate mai sus.

Pentru rezolvarea problemelor din categoria optimizării structurii sortimentale se pot aplica în multe cazuri metodele programării matematice.

Stabilirea necesarului de resurse corespunzînd unui plan determinat se face, în cazul produselor cu structură arborescentă, utilizînd sistemul de programe pentru calculatoare, cum ar fi sistemul BOMP pe calculatorul IBM sau sistemul ORACLE pentru calculatorul FELIX.

Problemele de scheduling sînt atît de diverse încît trebuie studiate metodele în număr considerabil, pe care le pune la dispoziție literatura de specialitate.

În practică, s-a constatat însă că metodele ADC-resurse, au aplicabilitate în multe tipuri particulare de probleme de scheduling.

Pentru fiecare din celelalte tipuri particulare de probleme menționate la începutul paragrafului, este necesară consultarea lucrărilor de specialitate.

Se înțelege că analiza de sistem prealabilă este absolut obligatorie pentru rezolvarea satisfăcătoare a problemelor de programare operativă a producției.

Scanned with OKEN Scanner

Din multitudinea de probleme şi aspecte ce pot apărea în teoria şi practica programării operative a producţiei, prezentăm în continuare, pentru ilustrare, problema ritmurilor de fabricaţie.

9.2. Stabilirea ritmurilor de fabricaţie în condiţiile executării unui plan realizabil la nivelul resurselor efectiv disponibile

Aceasta este o situaţie frecvent întîlnită în practică, deoarece în timpul executării planului pot apărea unele abateri între cantitatea de resurse necesară la un moment dat şi cea efectiv disponibilă. Planul de producţie iniţial se va modifica în funcţie de aceste abateri. În final, planul iniţial se va modifica, rezultînd un plan actualizat, pe care le vom denumi plan realizabil. Pentru a elabora acest plan realizabil pornim de la următoarele date de intrare.

— planul de producţie propus iniţial pentru o anumită perioadă;

— stocurile de materiale, subansamble estimate pentru o anumită perioadă;

— estimarea forţei de muncă disponibilă pe locuri de muncă pe perioada luată în considerare.

Pe baza planului propus iniţial, cu ajutorul pachetului de programe BOMP sau ORACLE se determină necesarul de materiale şi forţă de muncă corespunzător acestui plan.

În urma unor comparaţii între necesar şi disponibilul estimat, diferenţiat pe resurse, pot apărea situaţii de deficit sau excedent. Cunoscînd aceste date, factorii de decizie vor lua măsuri suplimentare de procurare a resurselor deficitare sau de redistribuire a celor excedentare. În finalul unei astfel de analize, care se face pentru un număr redus de resurse, în special resursele cheie, se poate calcula un disponibil estimat definitiv de resurse pentru care se va calcula planul realizabil.

Algoritmul de calcul pentru planul realizabil este următorul:

Pasul 1: Calculul necesarului de resurse pentru un plan dat iniţial.

Pasul 2: Compararea necesarului rezultat din calcul cu disponibilul definitiv estimat pe perioada luată în considerare.

Pasul 3: Determinarea coeficientului de ponderare.

Notînd:

p_{ij} — ponderea stocului j pentru produsul i;

NEC_{ji} — necesar stoc resursă j pentru produsul i;

Q_i — cantitatea din produsul i planificată a se fabrica în perioada de plan;

rezultă:

$$p_{ji} = \frac{NEC_{ji} \cdot Q_j}{\sum_i NEC_{ji} \cdot Q_i}$$

Scanned with OKEN Scanner

Pasul 4: Distribuirea stocurilor disponibile (mai mici decît necesarul) în funcție de un coeficient agregat de pondere (λ_i) care ia valori reale. Notînd :

S_j — stocul existent (dat în lista întocmită de factorii de decizie)

S_{ji} — stocul din resursa j pentru produsul i

$$S_j = \sum_i S_{ji}$$

rezultă :

$$S_{ji} = \frac{NEC_{ji} \cdot Q_i \cdot \lambda_i}{\sum_i NEC_{ji} \cdot Q_i} \cdot S_j.$$

Pasul 5: Determinarea numărului (Q_i') de produse finite ce se pot executa cu stocurile existente în ordinea priorității.

$$Q_i' = \min \left[Q_i, \frac{S_{1i}}{NEC_{1i}} ; \frac{S_{2i}}{NEC_{2i}} ; \ldots ; \frac{S_{ji}}{NEC_{ji}} ; \frac{F_A}{C_{iA}} \ldots \right]$$

în care :

F_{A1}, F_B ... reprezintă disponibilul de forță de muncă pe locuri de muncă ;

C_{iA}, C_{iB} ... — timpul necesar pe un anumit loc de muncă pentru executarea subansamblelor, semifabricatelor, care sînt cuprinse în produsul i.

Pasul 6: Calculul stocului rămas (S_{ji}'') :

$$Q_i' \times NEC_{ji} = S_{ji}' ; \quad S_{ji} - S_{ji}' = S_{ji}''.$$

Pasul 7: Se redistribuie stocul S_{ji}'', după care se reia de la pasul 5.

În funcție de cantitatea de produse/săptămînă ce rezultă din aceste calcule, șeful secției va stabili ritmul de produse semifabricate, piese, pe unitatea de timp dorită (zi, schimb).

Cînd numărul produselor și resurselor nu este foarte mare, problema poate fi rezolvată prin programarea liniară, bineînțeles utilizînd tehnica electronică de calcul.

9.3. Stabilirea ritmurilor de fabricație cu ajutorul tehnicilor de dinamică industrială (de tip Forrester)

Deciziile privind programarea operativă a producției, atît din punct de vedere static cît și dinamic pot fi perfecționate cu ajutorul tehnicilor de dinamică industrială. Avantajele aplicării acestor tehnici sînt :

— luarea în considerație a aspectelor dinamice ale desfășurării procesului de producție ;

Scanned with OKEN Scanner

— posibilitatea considerării oricăror alte legături cu diverse secții ale întreprinderii și eventual cu alte întreprinderi;

— posibilitatea evidenței diverșilor factori perturbatori pe o anumită perioadă de timp (lipsă de forță de muncă, lipsă materiale, lipsă semifabricate etc.);

— verificarea strategiilor propuse și alegerea unor strategii suboptime (privind programarea operativă a producției);

— analiza stabilității sistemului producției în timp etc.

Trebuie totuși menționat faptul că aplicarea tehnicilor Forrester prezintă și unele dezavantaje, dintre care cele mai importante sînt:

— dificultăți mari în stabilirea unei unități comune de măsură a producției realizate la diverse locuri de muncă sau secții;

— probabilitatea ridicată de a comite erori în urma alegerii unor unități de măsură convenționale;

— durata relativ mare de pregătire a datelor și de calcul în cazul cînd se urmărește o analiză în unități naturale sau mai apropiate de cele naturale;

— necesitatea însușirii tehnicilor de dinamică industrială de către personalul de conducere al întreprinderii.

Toate aceste dezavantaje pot fi însă înlăturate treptat, pe măsura aplicării metodei. Avantajul esențial al aplicării tehnicilor Forrester constă în posibilitatea de a stabili evoluția stării sistemului producției, toate variabilele modelului elaborat fiind funcții de timp.

La momentul inițial $\mathbf{T} = 1$ (ziua întîi a perioadei de plan) se consideră că toate variabilele sînt cunoscute. Perioada de timp pentru care se poate aplica simularea cu ajutorul tehnicilor Forrester este luna de plan. S-a convenit ca intervalul de timp să fie o zi ($\mathbf{T} = 1$ zi). A rezultat astfel că trebuie să se efectueze 25—26 cicluri de simulare, echivalentul zilelor efectiv lucrate.

Comportarea dinamică a procesului tehnologic modelat poate fi descrisă cu ajutorul unui set de ecuații (ecuații de stare ale modelului), format din *ecuații de nivel*, *ecuații de ritm* și *ecuații auxiliare*.

A. *Ecuațiile de nivel* permit stabilirea stocurilor în fiecare zi. Forma generală a unei ecuații de nivel este:

$$S^j(t) = S^j(t-1) + \Sigma\ [\mathbf{R}^{\mathbf{I}}(t-1) - \mathbf{R}^{\mathbf{E}}(t-1)] \cdot \Delta\,\mathbf{T}$$

în care $\Delta\,\mathbf{T}$ reprezintă intervalul de timp la care se efectuează simularea cu ajutorul tehnicilor Forrester (1 zi);

$S^j(t-1), S^j(t)$ — stocul existent la momentul $t-1$, respectiv momentul t, pentru resursa j;

$\mathbf{R}^{\mathbf{I}}(t-1)$ — ritmul intrărilor la momentul $t-1$;

$\mathbf{R}^{\mathbf{E}}(t-1)$ — ritmul ieșirilor la momentul $t-1$.

B. *Ecuațiile de ritm* se utilizează pentru determinarea cantităților ce se realizează în fiecare zi, în funcție de disponibilul de resurse. În general, ecuațiile de ritm sînt de forma:

$$S^j(t) = \min\left[\min_i\left[\frac{S^{ij}(t)}{C_s^{ij} n_s^j \mathbf{FT}_j}\right];\ \frac{\mathbf{FMD}_j(t)}{\mathbf{CFM}_j(t)};\quad \alpha_j(t)\,\frac{Q_{rj}^{(t)}}{TD_{rj}^{(t)}}\right]$$

Scanned with OKEN Scanner

în care :

$S^{ij}(t)$ reprezintă stocul din resursa i, repartizată la locul de muncă j;

C_s^{ij} — consumul specific la locul de muncă j pentru resursa i;

FT_j — fondul de timp pe schimb la locul de muncă j;

n_s^j — numărul de schimburi la locul de muncă j;

$FMD_j(t)$ — fondul de timp disponibil la locul de muncă j;

$CFM_j(t)$ — consumul de forță de muncă la locul j, pentru a produce o unitate din semifabricatul j;

$Q_{rj}(t)$ — cantitatea rămasă de executat la momentul t, calculată cu ajutorul relației :

$$Q_{rj}(t) - Q_{pj}(T) - Q_{ej}(t)$$

unde :

$Q_{ej}(t)$ reprezintă cantitate efectiv executată din reperul j pînă la momentul t;

$Q_{pj}^{(T)}(T)$ — cantitate planificată din reperul j pînă la momentul T (în care T este orizontul planificat, de exemplu, 26 zile lucrătoare);

$TD_{rj}^{(t)}$ — timpul disponibil rămas pentru j la momentul t;

$x_j(t)$ — coeficientul dinamic.

C. *Ecuațiile auxiliare* permit generarea unor variabile aleatoare (cum este cazul forței de muncă disponibile), generarea unor traiectorii aleatoare (cum este cazul coeficientului dinamic $x_j(T)$) în timp, generarea unor strategii condiționate, stabilirea valorilor funcției de eficiență economică etc.

Astfel, pentru forța de muncă disponibilă se generează o variabilă aleatoare uniformă (sau normală dacă este cazul) între două limite stabilite statistic.

În cazul coeficientului dinamic $x_j(t)$ se pot stabili funcții (liniare, exponențiale etc.), iar apoi pentru fiecare caz $t = t_i (i = 1, 2, \ldots, T)$ se adaugă sau se scade un reziduu aleator (cuprins între două limite date, de exemplu *0,01 — 0,3*).

***

Cele două exemple prezentate mai sus au doar rolul de a exemplifica modul de a aborda una din numeroasele probleme ale programării operative a producției.

Spațiul editorial limitat ne obligă să ne mărginim la aceste considerente și exemplificări generale privind programarea operativă a producției. Considerăm necesar să precizăm că, mai mult poate decît la oricare alta din problemele cercetării operaționale, în problema programării operative a producției este necesar un studiu teoretic și practic aprofundat. O bibliografie cuprinzătoare a acestei probleme poate fi consultată în [14].

Scanned with OKEN Scanner

BIBLIOGRAFIE

1. * * * — *Programul Partidului Comunist Român de făurire a societății socialiste multilateral dezvoltate și înaintare a României spre comunism*, Editura politică, București, 1975.

2. * * * — *Programul — directivă de cercetare științifică, dezvoltare tehnologică și de introducere a progresului tehnic în perioade 1981 — 1990 și direcțiile principale pînă în anul 2000*

3. Nicolae Ceaușescu — *Expunere la deschiderea colocviului privind problemele științei conducerii societății*, Editura politică, București, 1972.

4. Nicolae Ceaușescu — *Raport la Conferința Națională a Partidului Comunist Român, 7—9 decembrie 1977*, Editura politică, București, 1977.

5. Ackoff R. L., Sasieni M. W., — *Bazele cercetării operaționale*, Editura tehnică, București, 1975.

6. Aurian J., Boldur Gh., Lazăr S., — *Cercetarea operațională în construcții*, Editura științifică, București, 1967.

7. Bellman R. E., Dreyfus S. E., — *Programarea dinamică aplicată*, Editura tehnică, București, 1967.

8. Berge C., — *Teoria grafurilor și aplicațiile ei*, Editura tehnică 1969.

9. Boldur Gh., Ciobanu Gh., Băncilă I., — *Cartea Analistului de sisteme*, Editura științifică și enciclopedică, București, 1976.

10. Boldur Gh., — *Fundamentarea complexă a procesului decizional economic*, Editura științifică, București, 1973.

11. Boldur Gh., Stancu Minasian I., — *Programarea liniară cu mai multe funcții obiectiv : privire de ansamblu*, în „Studii și cercetări matematice" nr. 8, București, 1972.

12. Boldur Gh., Rațiu-Suciu C., Ciobanu Gh., Stancu Minasian I., — *Cercetarea operațională cu aplicații în economie*, partea I, curs litografiat, ASE, București, 1975.

13. Churchmann C. W., Ackoff L., Arnoff E., — *Introduction à la recherche operationnelle*, Dunod, Paris, 1970.

14. Ciobanu Gh., — *Algoritmi de ordonanțare și aplicații*, Teză de doctorat, Universitatea București, 1978.

Scanned with OKEN Scanner

15. Constantinescu P., Negoiță C. V., — *Sistemele informatice, modele ale conducerii și sistemelor conduse*, Editura tehnică, București, 1975.

16. Dantzig G. B., — *Linear Programming and Extension*, Princeton University Press, New York, 1963.

17. D'Epenoux F., — *Sur une problème de production et de stockage dans l'aleatoire*, R.R.O.* , Nr. 14. Paris, 1960.

18. Dinescu C., Postelnicu T., Săvulescu B., — *Matematici speciale aplicate în economie. Programare matematică și grafuri*. Editura didactică și pedagogică, București, 1977.

19. Ferrier J., — *La géstion scientifique des stocks*, Dunod, Paris, 1966.

20. Ghermeier B. I., — *Bazele teoretice ale cercetării operaționale*, Editura tehnică, București, 1974.

21. Hincin A. I., — *Matematiceskie metodi teorii massovovo obslujivania (Mathematical methods of the theory of mass service)* Trudi Mat. Inst. Steklov, vol. 49, Izdatelstvo Acad-nauk, USSR. Moscova, 1955.

22. Ionescu H., Mirescu P., Nădejde I., — *Programarea liniară*. Editura științifică, București, 1964.

23. Kantorovici L. V., — *Calculul economic al folosirii optime a resurselor*, Editura științifică, București, 1966.

24. Kaufmann A., — *Metode și modele ale cercetării operaționale, vol. I, II*, Editura științifică, 1967.

25. Kaufmann A., Cruon R., — *La programation dynamique et ses applications*, Dunod, Paris, 1965.

26. Lange O., — *Decizii optime*. Bazele programării, Editura științifică, București, 1970.

27. Lasdon L. S., — *Teoria optimizării sistemelor mari*. Editura tehnică, București, 1975.

28. Lazăr S., — *Analiza drumului critic*, Editura științifică, București, 1968.

29. Lazăr S., — *Alocarea și nivelarea resurselor*. Editura științifică, București, 1971.

30. Luce R. D., Raiffa H., — *Games and Decisions*. John Wiley & Sons, New York, 1958.

31. Maliță M., Zidăroiu C., — *Matematica organizării*, Editura tehnică, București, 1971.

32. Mănescu, Manea — *Rolul cibernetici în prognoza dezvoltării economico-sociale a României*, în Studii și cercetări de calcul economic și cibernetică economică, nr. 1, București, 1970.

33. Mănescu, Manea — *Modelarea economico-matematică*, în Studii și cercetări de calcul economic și cibernetică economică, nr. 3, București, 1970.

* Revue Française de Recherche Operationelle.

Scanned with OKEN Scanner

34. Mănescu, Manea Dumitru V., Ionescu V., Bărbatu Gh., — *Programarea matematică în industria petrolieră.* Editura Academiei, București, 1970.

35. Mănescu, Manea, Măcriș A., Dumitru, V., — *Metode moderne de organizare și planificare a lucrărilor de construcții-montaj.* Editura Academiei, București, 1967.

36. Mihăilă N., — *Introducere în programarea liniară.* Editura didactică și pedagogică, București, 1970.

37. Mihoc Gh., Ciucu Gh., — *Introducere în teoria așteptării.* Editura tehnică, București, 1967.

38. Mihoc Gh., Ciucu G., Mija A., — *Modele matematice ale așteptării.* Editura Academiei, București, 1973.

39. Mihoc Gh., Nădejde I., — *Programarea matematică vol. I + II.* Editura științifică, București, 1966, 1967.

40. Morse P. M., — *Queues, Inventories and Maintenance John Wiley & Sons*, New York, 1958.

41. Nădejde I., Bergthaller C., Zidăroiu C., Sburlan S., — *Problema de cercetare operațională*, Editura Academiei, București, 1971.

42. Onicescu O., — *Teoria jocurilor cu aplicații în programarea liniară*, Editura Academiei, București, 1962.

43. Rațiu-Suciu C., Boldur Gh., Ciobanu Gh., Dobre, I., — *Probleme și aplicații ale analizei drumului critic*, curs litografiat ASE, București, 1977.

44. Saaty T., — *Elements of Queueing theory with applications.* Mc Graw Hill, New York, 1961.

45. Săcuiu I., — *Cercetări operaționale în economie (Programarea dinamică).* Litografiat, A.S.E., București, 1973.

46. Simmonnard M., — *Programmation linéaire*, Dunod, Paris, 1962.

47. Teodorescu N., Boldur Gh., Stoica M., Stancu-Minasian I., Băncilă I., — *Metode ale cercetării operaționale în gestiunea întreprinderilor*, Editura tehnică, București, 1972.

48. Tovissi L., Oprescu Gh., — *Programarea liniară, partea I—V*, Curs litografiat A.S.E., București, 1974.

49. Tovissi L., Țigănescu E., — *Natura prețurilor umbră și folosirea lor în fundamentarea unor decizii economice*, în Studii de Cercetări de Calcul Economic și Cibernetică Economică, București, nr. 1, 1971.

50. Văduva I., Dinescu C., Săvulescu B., — *Metode și modele matematice în economie.* Editura didactică și pedagogică, București, 1974.

51. Zidăroiu C., — *Programarea dinamică discretă.* Editura tehnică, București, 1975.

Scanned with OKEN Scanner

Lei 11,30

Editura Didactică şi Pedagogică, Bucureşti – 1979

Scanned with OKEN Scanner